GF.G1

CCB-CCT-0002 D

E83-10225

TM-85-264

LGSOWG CCT FORMAT CCB DOCUMENT:

THE STANDARD CCT FAMILY OF TAPE FORMATS

```
(E83-10225)  LGSOWG CCT FORMAT CCB DOCUMENT:          N83-21466
THE STANDARD CCT FAMILY OF TAPE FORMATS
(NASA)   91 p HC A05/MF A01          CSCL 05B
                                                      Unclas
                                             G3/43    00225
```

AUGUST 28, 1979

TABLE OF CONTENTS

		Page
1.0	BACKGROUND	1-1
2.0	PURPOSE	2-1
3.0	SCOPE	3-1
4.0	APPLICABLE DOCUMENTS	4-1
5.0	THE CCT FAMILY FORMAT STRUCTURE	5-1
	5.1 SUPERSTRUCTURE CONCEPT	5-1
	5.2 SUPERSTRUCTURE RECORDS.	5-3
	5.3 BASIC CCT TAPE LAYOUT	5-5
	5.4 TAPE LAYOUT CONTINGENCIES	5-7
	5.4.1 Multi-volume Recording	5-7
	5.4.2 Adding Data to a Previously Recorded Tape	5-9
	5.4.3 Use of Text Records	5-9
6.0	LAYOUT OF SUPERSTRUCTURE RECORDS	6-1
	6.1 GENERAL RECORD FORMAT RULES AND CONTENT	6-1
	6.2 THE VOLUME DESCRIPTOR RECORD	6-5
	6.3 THE FILE POINTER RECORD	6-10
	6.4 FILE DESCRIPTOR RECORD	6-15
	6.5 THE TEXT RECORD	6-21
7.0	DESIGN STANDARD FOR FUTURE CCT FORMATS	7-1
	7.1 INTRODUCTION	7-1
	7.2 DATA GROUPING IN LOGICAL VOLUMES AND FILES	7-3
	7.2.1 Logical Volume	7-3
	7.2.2 File	7-3
	7.3 RECORDS	7-4
	7.3.1 Record Types	7-4
	7.3.2 Record Formatting	7-8

TABLE OF CONTENTS - (Continued)

7.4 DETERMINING FILE CLASS AND DEFINING THE FILE DESCRIPTOR VARIABLE SEGMENT 7-11

8.0 BRINGING EXISTING FORMATS INTO THE CCT FAMILY, AN EXAMPLE 8-1

8.1 THE LANDSAT-3 FORMAT (WITHOUT THE SUPERSTRUCTURE) 8-1

8.2 INCORPORATING THE SUPERSTRUCTURE 8-2

8.3 ASSIGNING FILE CLASS CODES AND VARIABLE SEGMENT FIELDS 8-9

APPENDIX A GLOSSARY A-1

APPENDIX B FILE DESCRIPTOR RECORD VARIABLE DATA SEGMENTS B-1

LIST OF FIGURES

Figure Page

5-1 EXAMPLE OF CONVERTING A PARTICULAR CCT FORMAT TO A CCT FAMILY FORMAT 5-2

5-2 BASIC CCT TAPE LAYOUT. 5-6

5-3 ILLUSTRATION OF CCT FAMILY TAPE LAYOUT CONVENTIONS 5-8

6-2 LAYOUT OF SUPERSTRUCTURE RECORDS 6-2

7-1 A STANDARD CCT RECORD. 7-9

8-1 LOGICAL VOLUME DATA FORMAT OF LANDSAT-3 CCTs (WITHOUT THE SUPERSTRUCTURE). 8-3

8-2 PLACEMENT OF EOFs. 8-4

9-3 EXAMPLE OF A SINGLE-VOLUME SET OF ONE BAND OF MSS IMAGERY BEFORE AND AFTER ADDING SUPERSTRUCTURE RECORDS 8-5

8-4 EXAMPLE OF A VOLUME SET OF FIVE BANDS OF MSS IMAGERY AFTER ADDING SUPERSTRUCTURE RECORDS. 8-7

LIST OF TABLES

Table Page

6.1 VOLUME DESCRIPTION RECORD 6-6

6.2 VOLUME DESCRIPTOR RECORD DATA FIELD EXPLANATIONS. 6-7

6.3 FILE POINTER RECORD 6-11

6.4 FILE POINTER RECORD DATA FIELDS EXPLANATICNS. 6-12

6.5 FILE DESCRIPTOR RECORD. 6-16

6.6 FILE DESCRIPTOR RECORD DATA FIELD EXPLANATIONS. 6-17

6.7 TEXT RECORD DATA. 6-22

6.8 TEXT RECORD DATA FIELD EXPLANATIONS 6-23

8.1 ADDITIONAL RECORDS AND TAPE FOOTAGE REQUIRED FOR
THE SUPERSTRUCTURE, BY VOLUME SET 8-8

8.2 THE LEADER FILE VARIABLE SEGMENT (LANDSAT-3). 8-10

8.3 THE LEADER FILE VARIABLE SEGMENT (LANDSAT-3) EXPLAINED 8-11

8.4 THE IMAGERY FILE VARIABLE SEGMENT (LANDSAT-3) 8-13

8.5 THE IMAGERY FILE VARIABLE SEGMENT (LANDSAT-3) EXPLAINED 8-15

8.6 THE TRAILER FILE VARIABLE SEGMENT (LANDSAT-3) 8-17

8.7 THE TRAILER FILE VARIABLE SEGMENT (LANDSAT-3) EXPLAINED 8-19

1.0 BACKGROUND

The Landsat-D CCT standards committee has been examining Computer Compatible Tape (CCT) fc.mats and format philosophies, with the objective of establishing some standard for CCTs which would promote information exchange among remote sensing data users and would allow data from a variety of sources to be used for a given application. Format requirements were collected from members and a draft document of these requirements was prepared by the Canada Centre for Remote Sensing (CCRS). This document was presented to the CCT standards committee meeting in June, 1978, at NASA Goddard Space Flight Center (GSFC). Also, at this meeting, F.E. Guertin (CCRS) presented a proposed standardization methodology which responded to these requirements, and which included an approach for bringing virtually all existing CCT formats into a CCT family of tape formats.

Key to the CCRS approach is a concept which, in this document, is referred to as a superstructure - a combination of precisely defined records and a method of employing them - which, when combined with any particular tape format, provides access to the data of that format without requiring specific knowledge of the particular format specifications. The concept won the general approval of the committee. It was decided that NASA, with the assistance of CCRS and Guertin, would produce an updated and expanded description of the superstructure and the CCT family of tape formats. The draft of this document was discussed and amended by the committee at a September 26 through 28, 1978 meeting at GSFC.

2.0 PURPOSE

The purpose of this document is to describe and define the tape format standardization approach recommended by the committee on CCT standardization. This approach applies to all types of remote sensing data user tapes. It should be noted that the purpose is to address user tapes as opposed to production tapes which may have additional system-imposed requirements which were not addressed by the committee.

3.0 SCOPE

The scope of this document is fourfold:

1. To present all rules and conventions required to employ the superstructure approach to the CCT family of tape formats for users of remote sensing data and producers of user tapes,

2. To specify the superstructure records,

3. To present the standard for future tape format design, which is a guide to designing the data records of a particular tape format, and

4. To provide an example of how to incorporate the superstructure into an already established tape format.

4.0 APPLICABLE DOCUMENTS

- "User Computer Compatible Tape Format Family Requirements." Dated: TBD; Document Number: CCB-CCT-0001 A
- "Recorded Magnetic Tape for Information Interchange (6250 CPI, group-coded recording)." Dated: 1976; Document Number: ANSI X3.54-1976.
- "Recorded Magnetic Tape for Information Interchange (1600 CPI, PE)." Dated: 1973; Document Number: ANSI X3.39-1973.
- "Interface Control Document between the Image Processing Facility and EDC Digital Image Processing System for Landsat for IPF/ EDIPS Computer Compatible Tapes (CCT's)." Dated: 6 June 1977; Document Number: IPF-ICD-116.
- "Format for Digital Imagery on Magnetic Tape." EIA Engineering Dept., Section 4.3.1. Dated: 18 January 1978; Document Number: Standard Proposal 1296.

5.0 THE CCT FAMILY FORMAT STRUCTURE

5.1 SUPERSTRUCTURE CONCEPT

There are presently many existing CCT formats for remote sensing data. To provide for interchangeability of these tapes among the users and producers of remote sensing data, a superstructure concept has been established. By adding the superstructure to any of the existing remote sensing data tape formats, the tapes will become part of the standard CCT format family and users will have the capability of performing basic processing (such as accessing and displaying) with the data without knowing the details of the individual tape formats.

The superstructure is composed of two basic components, a volume directory which globally describes the configuration of the tape or tape set and file descriptors which describe in more detail the configuration of the files. The files are logically grouped on a tape or set of tapes and this group is referred to as a logical volume. The individual tapes are the physical volumes. The volume directory introduces the logical volume and the file descriptor introduces the file (see Figure 5-1).

There are three types of records which comprise the superstructure: the volume descriptor, file pointer and file descriptor records, and the general structure of these records can be seen in Figure 6-1. Another record

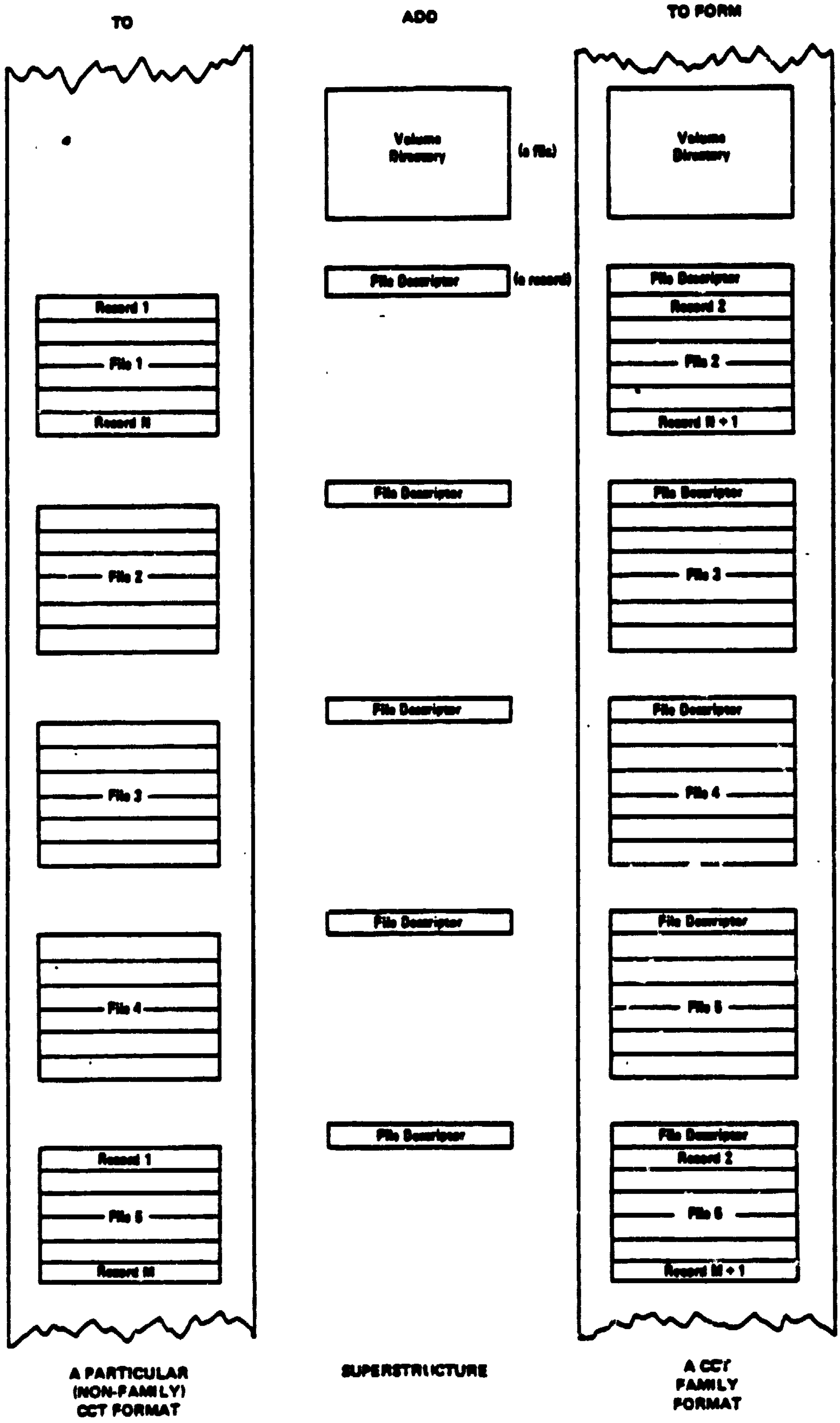


FIGURE 5-1. EXAMPLE OF CONVERTING A PARTICULAR CCT FORMAT TO A CCT FAMILY FORMAT

type is the text record which is used with the superstructure concept to provide any type of information in human readable form. The first 12 bytes are standard and appear on each type of record. They contain a record number, a record type code (which also includes sub-types) and a record length. The remainder of each record is dependent on record type. The volume descriptor and file pointer records each contain a field which is held free for utilization by the user.

For a complete description and definition of superstructure records, see Section 6.

5.2 SUPERSTRUCTURE RECORDS

In the volume directory file there is only one volume descriptor record and it is always the first record of the file (unless a text record preceeds it). It contains three general types of information: 1) the standard data, such as record number, type, length; 2) file-specific information, such as number of pointer records in the file; and 3) logical-volume-specific information. This third group of data is the most extensive and contains all the information which applies to the logical volume as a whole, such as data source identification, physical volume identification, and physical relationship of the logical volume to other logical volumes in the tape or tape set. (See Tables 6-1 and 6-2.) This record gives the user enough information to be able to locate the data in gross terms within the data tape set. Text records which contain additional information relating to the logical volume as a whole or in general may be present in the volume directory file.

The volume directory file also contains, for each of the remaining data files of the logical volume, a pointer record which points to the file and gives general information about the data in the file. The standard introductory data in the pointer record are followed by the identifying and descriptive information on the referenced file and its format. This includes file number, name, number of records, record lengths, and indication of the content of the file in terms of the type and format of the data. The file pointer records will allow a user to skip files and read only selected ones for performing rudimentary data processing.

The file descriptor record is the first record of each file of the logical volume (except the volume directory file) and it describes with more detail the data in the records of the file. The record contains the standard introductory information (e.g., record number, type, length) and information about the file (such as file number, name and file format) which will vary from file to file. It also has a segment which contains further identification and description of the file format and content; however, the data elements and layout of this segment depend on the class (type) of data within the file. This segment is called the variable data segment. For each file class there is defined a specific variable data segment. The file descriptor record gives a user enough information to access or display the data without requiring further specifications.

Text records begin with the standard introductory information (e.g., record number, type, length) and flags indicating the text is coded in ASCII or EBCDIC and whether the text continues on the next record. The remainder of the record can be used to write any desired alphanumeric information for any purpose. Text records can appear in any file. This type of record can be thought of as being similar to a Fortran comment card.

Each of the superstructure records contains a record sequence number, the record length and type code. The record sequence number is located in bytes 1 through 4 of each record and its value starts at 1 and increases sequentially in the subsequent records of the file. Bytes 9 through 12 of each record contain the record length. For the superstructure records, record lengths are: 360 bytes for the volume descriptor record; 360 bytes for the file pointer record; and the same length as the other records in the file for the file descriptor record if the records are of constant length within the file, or 360 bytes if the record lengths within the file are variable. The record type codes, which appear in bytes 5 through 8, are used to identify the type of information contained in the record.

There are four levels of type coding used by the superstructure, with one indicating the main type and the others indicating the subtypes. The main type codes and their data types are listed in Section 6.1.

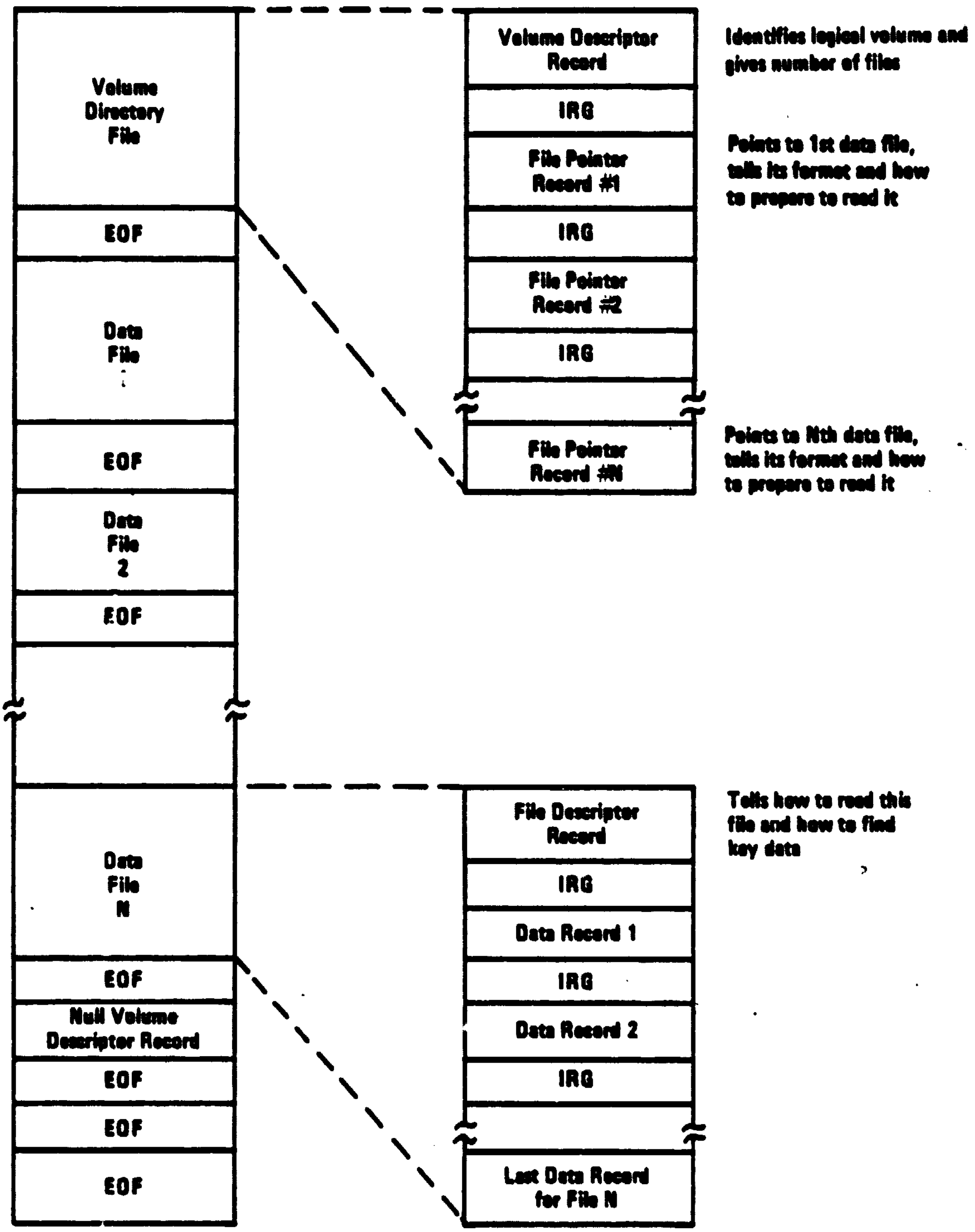
5.3 BASIC CCT TAPE LAYOUT

The simplest and most common form of CCT is the case where one physical volume (tape) contains one logical volume of data. A logical volume is a set of data which is grouped in any way that makes sense to the tape format designer (and presumably to the tape user). In terms of superstructure concepts, a logical volume is a set of data which is introduced by a volume directory file and concluded with a null volume descriptor record (or the volume directory file of a succeeding logical volume). It may contain one or more data files, each introduced by a file descriptor record.

The data files contain the actual information for which the CCT is recorded, while the superstructure records direct the user to this data. The layout of a CCT of one physical volume containing one logical volume of N data files is given in Figure 5.2. It starts with the volume directory file, which is the introduction to the logical volume and contains the volume descriptor and file pointer records. This is followed by the data files. The files are separated by end-of-file (EOF) indicators, and the records within a file are separated by inter-record gaps (IRGs). After the last data file, a null volume directory marks the end of the logical volume. It is a file consisting of a null volume descriptor record only.

If this particular tape (physical volume) is associated with other tapes so that together they form a set (referred to as a volume set), and if it is not the last volume of the set, the null volume descriptor record is not present and two EOFs indicate that there is no more data recorded on this tape. The two EOFs are referred to as an end-of-volume (EOV) indicator. If this particular tape is the last of a volume set or if it is a single-volume set (i.e., tape is not associated with other tapes as a set), the null volume descriptor record is followed by three EOFs, which is referred to as an end-of-set (EOS) indicator. (Systems which are unable to detect three consecutive EOFs will have to determine which logical volume is the last of a set by searching for the null volume descriptor record.)

ORIGINAL PAGE IS
OF POOR QUALITY



Identifies logical volume and gives number of files

Points to 1st data file, tells its format and how to prepare to read it

Points to Nth data file, tells its format and how to prepare to read it

Tells how to read this file and how to find key data

FIGURE 5-2. BASIC CCT TAPE LAYOUT

ORIGINAL PAGE IS
OF POOR QUALITY

5.4 TAPE LAYOUT CONTINGENCIES

Although recording one logical volume per physical volume is the simplest of tape formats, there are many situations which can make this inefficient or even impossible. A discussion of some of these situations will depict the tape layout conventions which apply.

5.4.1 Multi-Volume Recording

Multi-volume recording refers to recording a set of data which requires more than one physical volume. It generally implies that the volumes are recorded consecutively at a given time and site. The data can be recorded in the one logical volume per physical volume, as described, but when the length of the logical volume is unknown at recording start time, or if the logical volume is simply too long for one physical volume, the logical volume can be split between tapes. The logical volume may be divided between files, or when necessary between records within a file, although this second method is not recommended.

The method of splitting a logical volume on file boundaries is illustrated in the transition between physical Volumes 1 and 2 of Figure 5.3. The last file of Tape 1 is followed by two EOFs (an EOV). The first file of Tape 2 is the Volume Directory File. This is the same file which appeared in Tape 1, with the exception that certain data fields have been updated (e.g., Tape ID and Physical Volume Number). One of the fields to be updated is that indicating the number of the first data file of the present physical volume. When splitting the logical volume between the Nth and (N+1)th files, as in the illustration, this field would contain N+1 in the repeated volume directory. It is this field which indicates that this particular physical volume begins within a logical volume.

An example of splitting a logical volume within a file is illustrated between Physical Volumes 2 and 3 of Figure 5.3. The last record of Tape 2 is followed by two EOFs (an EOV). Tape 3 begins with the Volume Directory File - the same file which appears at the start of the logical volume on Tape 2, except that, once again, the proper data fields are updated.

ORIGINAL PAGE IS
OF POOR QUALITY

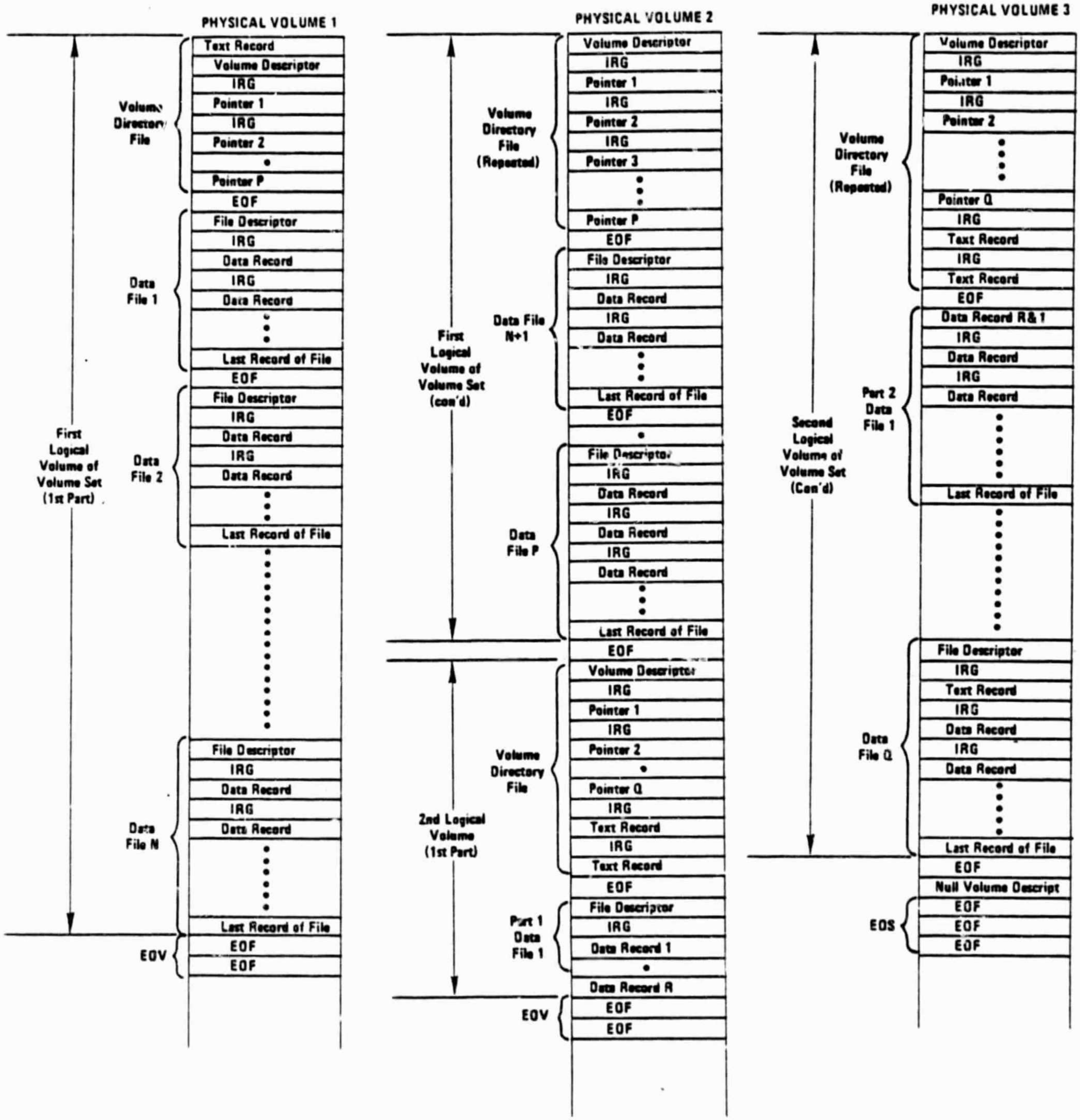


FIGURE 5-3. ILLUSTRATION OF CCT FAMILY TAPE LAYOUT CONVENTIONS

This includes a field in the file pointer record referring to the file being split and indicating the record number of the first record of that file on this tape. It is this field which indicates that this tape begins within a file. After an EOF the second portion of the split file is recorded without repeating the file descriptor record.

5.4.2 Adding Data to a Previously Recorded Tape

When adding data to a tape which already contains data, search for the null volume descriptor record, overwrite this record with a new volume descriptor record, add appropriate file pointer record(s), and then, after an EOF, record the new data files. Finally write a new null volume descriptor record followed by three EOFs. This procedure should be followed even when a small amount of data is to be added. Each time this procedure is followed a new logical volume is added to the tape.

5.4.3 Use of Text Records

The superstructure concept allows for text records to appear in any position in any file of a logical volume, although use of text records may be limited for particular applications (see Landsat-3 example, Section 9). There is one highly recommended use of text records. When a tape contains more than one logical volume, a text record, appearing as the first record of the first file of the tape, could contain a simple description of the logical volumes on the tape and thereby serve as a "tape directory."

Text records are of the same length as the other records of the file. If the other records of the file are not of equal length then text records are of 360 bytes in length.

6.0 LAYOUT OF SUPERSTRUCTURE RECORDS

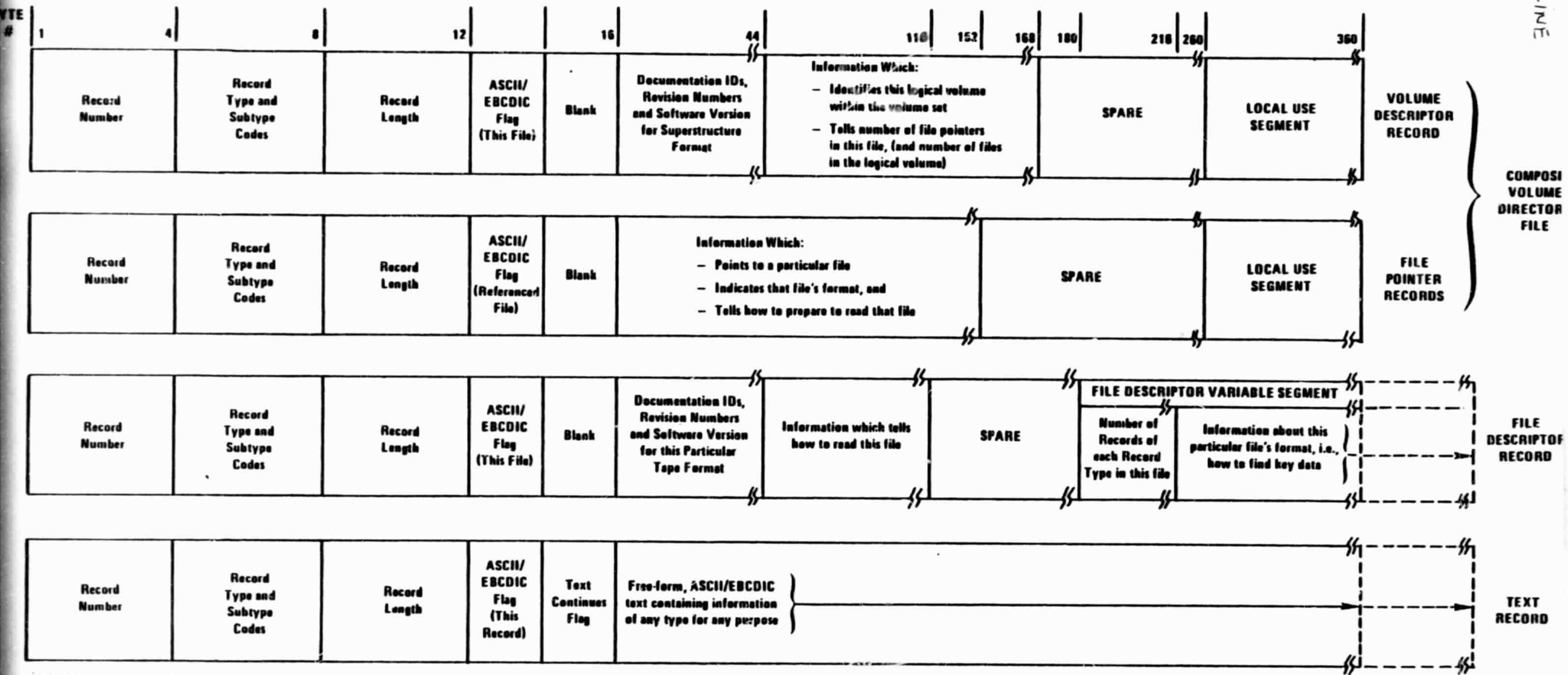
This section defines the format and content of the three superstructure records--volume descriptor, file pointer and file descriptor--and of the text record.

6.1 GENERAL RECORD FORMAT RULES AND CONTENT

The following rules apply to superstructure record format in general.

1. The first 12 bytes (6 fields) of all records contain only binary numbers and predefined bit-pattern codes.
2. The fields assigned to the first 16 bytes (8 fields) are similar for all four types of records.
3. From byte 13 to the end of record, fields are numeric or alphanumeric. They are coded in ASCII or EBCDIC, with ASCII being the preferred code.
4. Numeric strings are right-justified and alphanumeric are left-justified.
5. In fields containing data and blanks, the blanks are the character blank (ƀ) code in ASCII or EBCDIC.
6. Data fields are assigned so as to follow 4 byte boundary alignments.
7. Record lengths are a multiple of 180 bytes.

CUT → LINE



TE #
1
4
8
12
16
44
110
152
168
180
216
260
360
Record Number
Record Type and Subtype Codes
Record Length
ASCII/ EBCDIC Flag (This File)
Blank
Documentation IDs, Revision Numbers and Software Version for Superstructure Format
Information Which:
– Identifies this logical volume within the volume set
– Tells number of file pointers in this file, (and number of files in the logical volume)
SPARE
LOCAL USE SEGMENT
VOLUME DESCRIPTOR RECORD
Record Number
Record Type and Subtype Codes
Record Length
ASCII/ EBCDIC Flag (Referenced File)
Blank
Information Which:
– Points to a particular file
– Indicates that file's format, and
– Tells how to prepare to read that file
SPARE
LOCAL USE SEGMENT
FILE POINTER RECORDS
COMPOSI VOLUME DIRECTOR FILE
Record Number
Record Type and Subtype Codes
Record Length
ASCII/ EBCDIC Flag (This File)
Blank
Documentation IDs, Revision Numbers and Software Version for this Particular Tape Format
Information which tells how to read this file
SPARE
FILE DESCRIPTOR VARIABLE SEGMENT
Number of Records of each Record Type in this file
Information about this particular file's format, i.e., how to find key data
FILE DESCRIPTOF RECORD
Record Number
Record Type and Subtype Codes
Record Length
ASCII/ EBCDIC Flag (This Record)
Text Continues Flag
Free-form, ASCII/EBCDIC text containing information of any type for any purpose
TEXT RECORD

*File Descriptor and Text Records will be the same length (N) as the other records of the file. If the other records of the file are not of equal length, these records will then be 360 bytes in length.

FIGURE 6-1. LAYOUT OF SUPERSTRUCTURE RECORDS

ORIGINAL PAGE IS OF POOR QUALITY

CUT LINE

The 12 bytes referred to in rule 1 are illustrated in Figure 6-1. They contain record number, (1 field), record type and sub-type codes (4 fields), and record length (1 field). The record number is a 4 byte field which provides a binary sequential count of the records of a file. The first record of the file is numbered one and the record number increments by one per record. It is right justified with the left-most bit being the most significant as indicated in the following diagram:

Byte Number		1	2	3	4	
Bit Number	MSB	87654321	87654321	87654321	87654321	LSB
Bit Weight		2^{31}.....	2^{23}.....	2^{15}.....	2^{7}... 2^{0}	

The next 4 bytes are assigned to record type codes and sub-type codes. The type code is in byte 6. A code of 300)$_8$ in this byte indicates the record is one of the three superstructure records and a code of 077)$_8$ indicates a text record. The first sub-type code (hierarchically and in recording order) is in byte 5. In this byte, a code of 300)$_8$ indicates a volume descriptor record, a code of 333)$_8$ indicates a file pointer record, and a code of 077)$_8$ indicates a file descriptor record. The second sub-type code is in byte 7. In this byte a code of 077)$_8$ indicates a null volume descriptor record. All other sub-type code fields in superstructure records are coded 022)$_8$ at this time, which is the default code. These 4 bytes for the various records are thus coded as follows (all codes given in octal value):

Byte:	5	6	7	8
Field:	1st subtype code	type code	2nd subtype code	3rd subtype code
Volume descriptor record code:	300	300	022	022
Null volume descriptor record code:	300	300	077	022
File pointer record code:	333	300	022	022
File descriptor record code:	077	300	022	022
Text record code:	022	077	022	022

For a discussion on assigning record type codes for the various records of the data files, see Section 7.3.1.

Bytes 9 through 12 are reserved for the length of the record, given in bytes and coded in binary with bit weights as assigned to the four bytes for record number (above). For volume directory file records (volume descriptor and file pointer) this field is used for verification purposes only since these records are always 360 bytes in length. File descriptor and text records, however, are sized to equal the length of the other records of the file in which they appear (assuming the file is of constant record length; but if the records of the file are of differing lengths, the file descriptor and text records will be 360 bytes long).

The similarity of the next four bytes among superstructure records (rule 2) can be seen in Figure 6-1. The first two of these bytes (bytes 13 and 14) are ASCII/EBCDIC flags. The next two (bytes 15 and 16) are blanks, except when used to indicate the continuation of text on a following text record. These fields will be described on a per-record basis in the sections which follow.

Rule 7, that record lengths are multiples of 180 bytes, does have an exception. If the file descriptor or text records occur in a file of records whose length is not a multiple of 180 bytes, these records will be the same length as the records in the file, rather than a multiple of 180 bytes.

There is a similarity among the three superstructure records in content as well as in format. The purpose of these records are to identify, describe and locate data in the data files. Thus, there are fields such as tape identification (ID), logical volume generating country, and file name (identifying); number of pointer records in volume directory, file class, and file data type (describing); and physical volume number of start of logical volume, file number, and record code field location (locating). Thus superstructure records primarily supply information about the data on the CCT, rather than carrying data themselves. All the fields of these records are defined in the following sections.

6.2 THE VOLUME DESCRIPTOR RECORD

The volume descriptor is the first record of the volume directory file (unless preceded by text records). Its basic layout and content are illustrated in Figure 6-1. After the first 16 bytes of general information, the remainder of the record is composed of four segments. The first gives format documentation and software identification for the format in which the superstructure is recorded on this tape. The second segment provides basic information about the logical volume and gives the number of pointer records in the volume directory file. Since there is one pointer record for each file in the logical volume, this also gives the number of files. The third segment is spare, which is reserved for expansion of control information in future volume descriptor revisions. The fourth segment, the local use segment, provides space for whatever notation or information the tape user wants to carry with the volume directory. The individual data items of the volume descriptor record are listed in Table 6.1 and explained in Table 6.2.

TABLE 6.1
VOLUME DESCRIPTOR RECORD

FIELD		SEGMENT	BYTE#	DESCRIPTION
Type[1]	Number			
B	1		1-4	Record number
B	2		5	1st record subtype code, always 300)$_8$ = volume directory
B	3		6	Record type code, always = 300)$_8$ = superstructure
B	4		7	2nd record sub-type code = 077)$_8$ if null volume descriptor, otherwise this sub-type code = 022)$_8$
B	5		8	3rd record sub-type code, always = 022)$_8$
B	6		9-12	Length of this record
A	7		13-14	ASCII/EBCDIC Flag for this file
	8		15-16	Blank
A	9	1	17-28	Superstructure control document number
A	10		29-30	Superstructure control document revision number
A	11		31-32	Superstructure record format revision letter
A	12		33-44	Software release number
A	13	2	45-60 **	ID for physical volume containing this volume descriptor (tape ID)
A	14		61-76 *	Logical volume ID
A	15		77-92	Volume set ID
N	16		93-94	Number of physical volumes in the set
N	17		95-96	Physical volume number, start of logical volume
N	18		97-98	Physical volume number, end of logical volume
N	19		99-100**	Physical volume number containing this volume descriptor
N	20		101-104**	First referenced file number in this physical volume
N	21		105-108	Logical volume number within volume set
N	22		109-112**	Logical volume number within physical volume
A	23		113-120*	Logical volume creation date
A	24		121-128*	Logical volume creation time
A	25		129-140*	Logical volume generating country
A	26		141-148*	Logical volume generating agency
A	27		149-160*	Logical volume generating facility
N	28		161-164*	Number of pointer records in volume directory
N	29		165-168*	Number of records in volume directory
	30	3	169-260	Volume descriptor spare segment
	31	4	261-360	Local use segment

* Undefined in null volume descriptor.

** Fields to be updated in a repeated volume directory.

[1] B = binary, A = alnhanumeric, N = numeric

TABLE 6.2

VOLUME DESCRIPTOR RECORD
DATA FIELD EXPLANATIONS

FIELDS	EXPLANATIONS
1 thru 6	As described in Section 6.1.
7	The ASCII/EBCDIC flag indicates if the alphanumeric information in the volume directory file is in ASCII or EBCDIC. A code of Ab̸ denotes ASCII and Eb̸ indicates EBCDIC.
8	Two blanks.
9	12 characters giving the Superstructure Format Control Document identifying number, i.e., the number of that document which defines the current superstructure record formats and conventions - this document.
10	2 characters indicating the revision number or letter of the Superstructure Format Control Document, i.e., this document.
11	2 characters indicating the revision letter of the superstructure record formats. Initially coded b̸A, this code updates one letter character, alphabetically, each time there is a change to the format of a superstructure record (as opposed to a change to the control document which may not have been a change in actual record format). The 26th revision is coded AA, the 27th AB, the 28th AC, and so on.
12	12 characters identifying the software version. The software referred to here is that used to write this logical volume. The code is alphanumeric, left-justified code assigned by the producing facility. It is updated for each modification.
13	This is a 16 character code also written or printed externally on the physical volume and used to uniquely reference a particular CCT. This identification is the same for all logical volumes on the same physical volume. When a logical volume spans physical volumes the code is updated for the continuation physical volume(s). (Also referred to as tape ID.)
14	This is a 16 character code supplied at the time the logical volume is recorded and which can be used to uniquely reference a logical volume within a volume set.

TABLE 6.2

(CONT'D)

FIELDS	EXPLANATIONS
15	This is a 16 character code supplied at the time the first physical volume of a volume set is recorded. It ensures a unique way to reference a volume set consisting of multiple physical volumes.
16	This indicates the total number of physical volumes in a volume set. A blank field indicates that the information is not available at the time the logical volume is recorded.
17	This indicates the sequence number of the physical volume, within a volume set, which contains the 1st record of the logical volume, A blank field implies no specific sequence. The first physical volume is numbered as 1.
18	This will be the same as above unless the logical volume is split across physical volumes. It indicates the sequence within a volume set of the physical volume containing the last record of the logical volume. It should be coded blank if unknown at time of recording.
19	This is the sequence number within the volume set of the physical volume containing this volume directory file. If the logical volume is completely contained within one physical volume this field will be coded the same as field 17. If it spans physical volumes, the volume directory is repeated at the beginning of each tape containing part of the logical volume, and this field indicates the tape number of the current physical volume.
20	This field gives the file number within the logical volume of the first file which follows this volume directory. This can be larger than one (the number of the first data file of a logical volume) when a logical volume spans multiple physical volumes. If a file spans two or more physical volumes each portion of the file is referenced by the same number (because each portion is using the same volume pointer record). Volume directory files are not included in the file sequence number count.
21	This indicates the sequence number of the present logical volume within a volume set. Null volume descriptor is included in this count. The first logical volume is denoted as 1.

TABLE 6.2

(CONT'D)

FIELDS	EXPLANATIONS
22	This indicates the sequence number of the present logical volume within this physical volume. This number is always present even in a null volume descriptor. The first logical volume is denoted as 1. When a logical volume spans multiple physical volumes, the portion of the logical volume on this tape is counted here as if it were an entire logical volume. This rule does not apply to field 21 — logical volume number within volume set.
23	This is the date when the logical volume is recorded, expressed in years, months and days. If multiple logical volumes are recorded at different dates on the same physical volume, each logical volume will reflect its own creation date. The code is of the form: YYYYMMDD
24	This is the time when the logical volume is recorded, expressed in hours, minutes and seconds. Due to the time required to record a logical volume it is unlikely that two logical volumes will exhibit the same time. The code is of the form: HHMMSSXX--where XX indicates hundredths of seconds.
25	Name (or abbreviation) of country generating this logical volume.
26	This specifies the laboratory or center responsible for the creation of the logical volume.
27	This identifies the computer facility on which the logical volume is recorded.
28	The number of pointer records in this volume directory. (This automatically indicates the number of data files in the logical volume.)
29	Total number of records in this volume directory. Equals number of pointers plus number of text records plus 1.
30	This is the portion of the directory which is undefined and unused. It is reserved for subsequent revisions.
31	This is the portion of the directory which can be used locally without having to satisfy any common and approved standard. Its purpose is to meet local requirements that are not universally recognized.

6.3 THE FILE POINTER RECORD

File pointer records reside in the volume directory file. There is one file pointer record for each data file of the logical volume. These records are recorded in the same sequence as the files to which they point.

The general record format and content of the file pointer record are illustrated in Figure 6-1. After the first 16-bytes of general information, there are three data segments. The first segment supplies information which points to (locates) one particular data file, indicates that file's format, and tells how to prepare to read the file. The second segment is space and is reserved for expansion of the file pointer information segment in future format revisions. The third segment provides space which the tape user may use as desired. The individual fields of the file pointer record are listed in Table 6-3 and explained in Table 6-4.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 6.3
FILE POINTER RECORD

FIELD		SEG-MENT	BYTE #	DESCRIPTION
Type[1]	Number			
B	1		1-4	Record number
B	2		5	1st record sub-type code = 333)$_8$ = pointer
B	3		6	Record type code, always = 300)$_8$ = super-structure
B	4		7	2nd record sub-type code = 022)$_8$
B	5		8	3rd record sub-type code = 022)$_8$
B	6		9-12	Length of this record
A	7		13-14	ASCII/EBCDIC flag for the referenced file
	8		15-16	Blank
N	9	1	17-20	Referenced file number
A	10		21-36	Referenced file name
A	11		37-64	Referenced file class
A	12		65-68	Referenced file class code
A	13		69-96	Referenced file data type
A	14		97-100	Referenced file data type code
N	15		101-108	Number of records in referenced file
N	16		109-116	Referenced file 1st record length
N	17		117-124	Referenced file maximum record length
A	18		125-136	Referenced file record length type
A	19		137-140	Referenced file record length type code
N	20		141-142	Referenced file physical volume number, start of file
N	21		143-144	Referenced file physical volume number, end of file
N	22		145-152***	Referenced file portion, 1st record number for this physical volume
	23	2	153-260	Pointer spare segment
	24	3	261-360	Local use segment

*** Updated in repeated volume directory if logical volume is split within a file.

[1] B = binary, A = alphanumeric, N = numeric

TABLE 6.4

FILE POINTER RECORD
DATA FIELD EXPLANATIONS

FIELD	EXPLANATION
1 thru 6	As described in section 6.1.
7	A 2-byte flag indicating whether the alphanumeric data in the referenced file is coded ASCII or EBCDIC. If ASCII, this field is coded Ab̸; if EBCDIC, it is coded Eb̸.
8	Two blanks
9	Sequence number within the logical volume of the file referenced by this pointer. The first file following the volume descriptor is file number 1.
10	A 16 character name which is the unique identification provided when the volume directory is created in order to specify the file referenced by the pointer. The name indicates the nature of the file: header, annotation, imagery, etc.
11	This is the description of the class to which the referenced file belongs. The class of a file is based on the nature of its content. The number of classes should be open-ended but limited. Classes which are essential are: - Alphanumeric and numerical lists - Cellular or image data - Profile data - Polygon data - Isolated data points. A file class indicates a particular file format, and hence, knowledge of the class of a file leads directly to knowledge of that file's format. Each file class has a class code associated with it which is given in the next field.
12	A 4-byte code for the class described in field 11.

TABLE 6.4 (Cont.)

FILE POINTER RECORD
DATA FIELD EXPLANATIONS

FIELD **EXPLANATION**

13 This field identified only the data; not the record introduction (even if it is not stored in binary or in the first twelve bytes) contained in the file through use of the following phrases:

- 8 BIT ASCII ONLY
- EBCDIC ONLY
- BCD ONLY
- BINARY ONLY
- MIXED BINARY AND ASCII
- MIXED BINARY AND EBCDIC
- MIXED BINARY AND BCD
- UNDEFINED, ETC.

Each data type has a code associated with it, which is given in the following field (field 14).

14 A 4-byte code for the data type described in field 13. These codes (given in the order of the phrases above) are: ASCO, EBCO, BCDO, BINO, MBAA, MBAE, MBAB, UNDF.

15 This indicates the number of records in the referenced file. If this number is not known at the creation time, then this field is blank.

16 The length, in bytes, of the file descriptor record in the referenced file.

17 This is the length in bytes of the longest record in the referenced file other than the file descriptor record. This is necessary to determine the memory requirement before accessing the file itself.

18 The types are: fixed length, variable length, undefined length, and length defined in the file descriptor. They are written as: FIXED LENGTH, VARIABLE LEN, UNDEFINED LE, and LENGTH IN FD. The size of fixed length records is given by field 17. Variable length records are smaller than the maximum size described above and the actual size of each record is defined in a fixed location in the record itself. Undefined length records are smaller than the maximum size and the exact length is not given in the format definition. It has to be obtained from other sources. For some file classes, variable length records have lengths defined in the file descriptor record.

TABLE 6.4 (Cont.)

FILE POINTER RECORD
DATA FIELD EXPLANATIONS

FIELD	EXPLANATION
19	A 4-byte code for the type described by field 18. The codes are: FIXD, VARE, UNDD, and LIFD.
20	The number of the physical volume within the physical volume set containing the first record of the file. May be left blank if information unknown at the time of recording.
21	The number of the physical volume within the physical volume set containing the last record of the file. May be left blank if information unknown at time of recording.
22***	When a portion of the referenced file is on the previous physical volume, this number is the record number of the first record of the referenced file to be recorded on this physical volume. In all other conditions the value is one.
23	This is the portion of the pointer record which is undefined and unused. It is reserved for subsequent revisions.
24	This is the portion of the pointer record which can be used locally without having to satisfy any common and approved standard. Its purpose is to meet local requirements that are not universally recognized.

*** Undated in repeated volume directory if the logical volume is split within a data file.

6.4 FILE DESCRIPTOR RECORD

A file descriptor record introduces each data file. The general record format and content of file descriptor records are illustrated in Figure 6-1. Following the first 16-bytes of general information are four record segments. The first segment identifies the documentation of the format of, and the software version used to produce, the data file of the particular application with which the superstructure is being used. In other words, while the segment comparable to this in the volume descriptor identifies current documentation of the superstructure formats, this field identifies current documentation of the formats of the remainder of the records.

The second segment provides the basic information necessary to read this file (the file containing this file descriptor record.) The third segment is the spare which is reserved for expansion in future file descriptor revisions.

The fourth segment is referred to as the file descriptor variable segment. This is because this segment varies with file class. Just as a particular file class indicates a particular file format, it also implies a particular file descriptor variable segment. The variable segment for a particular file format is chosen from among existing variable segments if any apply, or else it is defined at the time that the particular format is designed. The layouts of variable segments which have already been defined are given in Appendix B. These segments commonly start with parameters indicating the number of records of each record type in the file. This is followed with locator information particular to the format of the data file, i.e., how to access and display essential data.

The data fields of the file descriptor record (other than those of the variable segment) are listed in Table 6.5 and explained in Table 6.6.

CUT OFF

FILE DESCRIPTOR RECORD

FIELD		SEGMENT	BYTE #	DESCRIPTION
Type 1/	Number			
B	1		1-4	Record number
B	2		5	1st record sub-type code = $077)_8$ = file descriptor
B	3		6	Record type code, = $300)_8$ = super-structure
B	4		7	2nd record sub-type code = $022)_8$
B	5		8	3rd record sub-type code = $022)_8$
B	6		9-12	Length of this record
A	7		13-14	ASCII/EBCDIC flag for this file
	8		15-16	Blanks
A	9	1	17-28	Control document number for this embodiement
A	10		29-30	Control document number for this embodiement revision number
A	11		31-32	File design descriptor revision letter
A	12		33-44	Software release number
N	13	2	45-48	File number
A	14		49-64	File name
A	15		65-68	Record sequence and location type flag
N	16		69-76	Sequence number location
N	17		77-80	Sequence number field length
A	18		81-84	Record code and location type flag
N	19		85-92	Record code location
N	20		93-96	Record code field length
A	21		97-100	Record length and location type flag
N	22		101-108	Record length location
N	23		109-112	Record length field length
A	24		113	Flag indicating whether information resulting from image analysis is included within the variable segment of this record
A	25		114	Flag indicating whether information resulting from image analysis is included within this file
A	26		115	Flag indicating that data display information is included within the file descriptor record
A	27		116	Flag indicating that data display information is included within the file in record(s) other than the file descriptor
	28	3	117-180	Reserved segment
	29	4	181-end-of- record*	File Descriptor Variable Segment

* Typically the file descriptor will be the same length as the remaining records of the file. If the file contains records of variable length, the file descriptor will be 360 bytes in length.

1/ [illegible]

CUT OFF

TABLE 6.6

FILE DESCRIPTOR RECORD
DATA FIELD EXPLANATIONS

FIELD	EXPLANATIONS
1 thru 6	As defined in section 6.1.
7	The ASCII/EBCDIC flag indicates if the alphanumeric data of this data file is in ASCII or EBCDIC. A code of Aƀ indicates ASCII; of Eƀ indicates EBCDIC.
8	Two blanks.
9	12 characters giving the control document number of that document which defines this file format.
10	2-bytes giving the revision number of the control document defining the current file format.
11	2-bytes giving the revision letter of the file format (as opposed to revisions which affect the control document without affecting the file format). For a description of the lettering scheme, see field 11 of the volume descriptor record, Table 6.2.
12	12 characters identifying the software version. The software referred to here is that used to write this file, (i.e., to write this data file).
13	Sequence number of this file within the logical volume. The volume directory file is not included in this count.
14	This is the unique 16-character identification of the present file as stated in the volume directory file.
15	This is the flag which indicates whether each record in the file has a sequence number, if the location is fixed or variable, or if the count is cyclical. The file descriptor itself always has a sequence number. It is not required for the other records. The allowed codes and their meanings are as follows: NSEQ-no known record sequence numbers present in the data records of the file. FSEQ-the record sequence number is present in the same location in all data records of the file.

TABLE 6.6

FILE DESCRIPTOR RECORD
DATA FIELD EXPLANATIONS
(CONT'D)

FIELDS **EXPLANATIONS**

VSEQ-record sequence numbers are present in the data records of the file but their locations vary from record to record.

CSEQ-record sequence numbers are not present, but a cyclic counter is present, i.e., each group of X records excluding the file descriptor record in the file is numbered 1 through X.

If this field is coded NSEQ or VSEQ, fields 16 and 17 are blank. If it is coded FSEQ or CSEQ, these fields are coded as follows.

16 These eight bytes give the location of the start of the sequence number field. They give the record byte number of the first byte of the field. (It is assumed that the record sequence number field falls on a byte boundary and consists of an integral number of bytes.)

17 4-bytes indicating the length, in bytes, of the record sequence number field.

18 This flag indicates whether each record in the file has a record type code and if the location of the code is fixed or variable. The file descriptor itself always has a record code. It is not required for the other records. The allowed codes and their meanings are as follows:

NTYP-no known record type codes present in the data records of the file.

FTYP-the record type code is present in the same location in all the data records of the file.

VTYP-the record type codes are present in the data records of the file but their locations vary from record to record.

If this field is coded NTYP or VTYP, fields 19 and 20 are blank. If it is coded FTYP, these fields are coded as follows.

TABLE 6.6

FILE DESCRIPTOR RECORD
DATA FIELD EXPLANATIONS
(CONT'D)

<u>FIELDS</u> <u>EXPLANATIONS</u>

19 These 8-bytes give the location of the start of the record type code field. They give the record byte number of the first byte of the field. (It is assumed that the record type coae field falls on a byte boundary and consists of an integral number of bytes.)

20 4-bytes indicating the length, in bytes, of the record type code field.

21 This flag indicates whether each record of the file has its record length recorded within the record, and if the location of the code is fixed or variable. The file descriptor itself contains a record length field. This is not required for the other records. The allowed codes and their meanings are as follows:

NLGT-no known record lengths present in the data records of the file.

FLGT-the record length field is present in the same location in all the data records of the file.

VLGT-the record length fields are present in the data records of the file, but their locations vary from record to record.

If this field is coded NLGT or VLGT, fields 22 and 23 are blank. If it is coded FLGT, these fields are coded as follows.

22 These 8-bytes give the location of the start of the record length field. They give the record byte number of the first byte of the field. (It is assumed that the record length field falls on a byte boundary and consists of an integral number of bytes.)

23 4-bytes indicating the length, in bytes, of the record length field.

24 This flag indicates whether information resulting from image analysis is included within the variable segment of this record. The code for yes=Y, no=N.

TABLE 6.6

FILE DESCRIPTOR RECORD
DATA FIELD EXPLANATIONS
(CONT'D)

FIELDS	EXPLANATIONS
24	This flag indicates whether information resulting from image analysis is included within the file itself. The code for yes = Y, no = N.
26	This flag indicates whether information necessary to display the file is included within the variable segment of this record. The code for yes=Y, no=N.
27	This flag indicates whether information necessary to display the file is included within the file itself. The code for yes=Y, no=N.
28	60-bytes which are held in reserve for expansion of file descriptor information in future format revisions.
29	The file descriptor variable segment.

6.5 THE TEXT RECORD

While the text record is not specifically a superstructure record, the superstructure concept provides for the use of text records. The format of text records is very simple, as illustrated by Figure 6-1. After the first 16-bytes of basic introductory data, the remainder of the record is free-form text. It may be used to carry any type of information for any purpose, as long as it is in alphanumeric code. It may appear in any position in any file (although this freedom of placement may have to be limited somewhat for particular file formats-see section 8 for an example). In its ability to float and to carry information which is not pre-defined, the text record is analogous to the "comment" card used in Fortran programming.

The data fields of the text record are listed in Table 6.7 and explained in Table 6.8.

The length of the text record obeys the following rules: in a fixed record length file it has the same length as the other records in the file; in a variable record length file it is 360 bytes long.

The text records can be used only in files for which the first 12 bytes (6 fields) of every record are compatible with the text record's first 12 bytes.

TABLE 6.7
TEXT RECORD DATA

FIELD		BYTE #	DESCRIPTION
TYPE*	NUMBER		
B	1	1-4	Record number
B	2	5	1st record sub-type code = $022)_8$
B	3	6	Record type code = $077)_8$ = text
B	4	7	2nd record sub-type code = $022)_8$
B	5	8	3rd record sub-type code = $022)_8$
B	6	9-12	Length of this record
A	7	13-14	ASCII/EDCDIC flag for this record
A	8	15-16	Continuation flag
A	9	17-end-of record	Field to be used for free-form text

*B = binary, A = alphanumeric

TABLE 6.8

TEXT RECORD DATA FIELD EXPLANATIONS

FIELDS	EXPLANATIONS
1 thru 6	As described in section 6.1.
7	A flag indicating whether the alphanumeric information of this record is coded ASCII or EBCDIC. A code of Ab̸ indicates ASCII; of Eb̸ indicates EBCDIC.
8	This field contains two blanks unless the information of this record is continued on a following text record. In this case, this field is coded Cb̸.
9	The remainder of this record is reserved for alphanumeric information of any type for any purpose desired by the tape producer or tape user. It is recommended that the text message end with at least one null character. (The null character is as defined for ASCII or EBCDIC.)

7.0 DESIGN STANDARD FOR FUTURE CCT FORMATS

7.1 INTRODUCTION

The CCT superstructure is rigidly formatted and precisely defined. It is also, of necessity, very general in the sense that it can be applied to a great variety of data/tape formats. It was designed in this manner in order to accommodate as many existing and future remote sensing data formats as possible. It is highly desirable, however, to approach as nearly as possible one universal CCT tape format to be used for all future applications.

Since future remote data sensing, manipulation and interpretation techniques are unknown, it is difficult to predict the form of future experiment data. It is unreasonable to expect a format, for which data records have been defined to the precision to which the superstructure records have been defined, to be appropriate for all types of data. The method for satisfying the conflict in this situation is twofold. First, define the superstructure which will allow for all the possibilities. Second, define a standard toward which all future particular tape formats will be designed. This standard would not state rules as to what is permitted in the design, but rather what is preferred. In this way an overall CCT universal tape format is encouraged, with particular formats converging to the standard.

The preceding sections of this document have defined the superstructure records, thus step one toward the solution is satisfied. This leaves step two: define a standard for the design of future CCT formats. What does this entail? Since CCTs will contain superstructure records and r'cords which we have referred to as data records, and since the superstructure records have already been defined, it is clear that such a standard would have to address the definition of the data records. The superstructure concept also establishes certain basic tape organization procedures, but it does not address the ordering of what have been referred to as data files, nor does it address the organization of records within a data file. Thus, the formatting of these data files should also be covered by the standard.

Designing what is referred to as a particular tape format can be considered to involve the following tasks:

1. Decide what grouping of data will compose a logical volume. This is the highest level design decision and is based solely on what would be a reasonable (logical) data grouping. It could include, for example: one scene of image data; one scene of image data plus one associated land use classification map; data from one experiment from sensor turn-on to sensor turn-off; etc.

2. Group data of the logical volume into files. This decision is also based on what would be a reasonable data grouping. A very important factor in grouping data into files is that the format of each resulting file should be one which will have as wide an application as possible, i.e., that the format is not of a nature to restrict it to only rare or occasional use.

3. Determine the types of records to be contained in, and their ordering in, each file. A prime consideration here is that a record of a given type contain primarily one type of data.

4. Define the specific record formats.

5. Determine the class of each file format and define a file descriptor record variable segment for each.

A standard for designing a tape format should provide a guide to these tasks. This guide is presented in the following sections.

7.2 DATA GROUPING IN LOGICAL VOLUMES AND FILES

7.2.1 Logical Volume

While sets of related data are grouped physically in tapes and tape sets, the highest order of logical data grouping addressed by a particular standard format is the logical volume. While one or more logical volumes in one or more particular formats may be contained in one or more physical volumes, within each logical volume only one particular format applies.

The relationship of logical volumes to one another on a tape or tape set, regardless of the data content of a logical volume, is completely defined by superstructure conventions as described in Section 5.4. Thus, when designing a format for a mission, project, sensor, etc., the logical volume is the highest order of data organization that need be considered. Definition of the data content of a logical volume, or of data grouping into logical volumes, is a design decision for each particular format which must be based almost entirely on the nature of the data for which the format is being designed. Because of this there is very little which can be offered as a guide other than that a logical volume of data should be a stand-alone data set, i.e., should include experiment data plus support and associated data required to make use of the data. This does not exclude the possibility of a logical volume of nothing but raw experiment data, if data in this form serve a purpose.

7.2.2 File

The internal structure of a logical volume is defined by the superstructure concept as defined in Section 5.3. It consists of a volume directory file (defined by the superstructure) and one or more data files. Once the data content of a logical volume has been determined, it then remains to group this data into data files. This grouping of data into files will establish the file format.

The files within a logical volume may vary from being all of one format to being all of differing formats. The order of file types within a logical volume is also to be established by the particular format.

A standard data file has a file descriptor as its first record, as defined by the superstructure, and one or more other records of one or more record types. If more than one other record type is present, the order in which the various record types appear is also to be defined by the file format. Each resulting file format must have an associated file descriptor record variable segment. When designing a particular format, previously defined file formats (and their associated file descriptor variable segments) should be used whenever possible. When it is necessary to design a new file format, it is desirable to include as few record types as practical.

7.3 RECORDS

Once the basic organization of each type of file has been established, it remains to determine the type of records appropriate to carry the data, and to define standard record formats.

7.3.1 Record Types

The concept of record type is especially important in records to be used in conjunction with the superstructure. All standard records have record type codes. When these codes are assigned in accordance with the standard guidelines, they will not only indicate the type of data within the record, but will also become a code for the specific format of the record. Record codes thus become the lowest level of, and most precise vehicle for, format definition. And since all data on any CCT are in records, proper use of record codes along with the other basic superstructure concepts can completely define the format of any CCT.

7.3.1.1 Record Content. A standard record should contain one basic type of data, although associated information may also be included. For example, a typical record of data from an earth scanning sensor may include one scan line of imagery plus calibration wedge values, a line quality code, a line identifying code, border fill pixels, or other scan line associated support information. This record is still considered an image data record. This method of segregating data of different types into different records is important since the nature of the basic data in the record is the initial factor in determining the record type code.

7.3.1.2 Record Type Coding. The record type code is actually made up of four separate one-byte codes, a type code and three sub-type codes. The type code indicates the basic data type of the record. The three sub-types are used to further classify the record data and format. Once a new record format has been designed, an appropriate four-code type code is assigned to it which will be unique to that format.

7.3.1.2.1 Basic Record Type. In assigning record type codes, the first step is to choose one of the eight basic data type codes based on data type. The data types and the codes associated with them are as follows:

DATA TYPE	CODE	DATA TYPE	CODE
Superstructure	300)$_8$	Ancillary	044)$_8$
Tape Directory	011)$_8$	Data	355)$_8$
Header	022)$_8$	Trailer	366)$_8$
Annotation	333)$_8$	Text	077)$_8$

In choosing the basic record type, the following guidelines should be followed:

1. Superstructure - this is used with records of the superstructure (volume descriptor, file pointer, and file descriptor) only, and therefore is not an appropriate code for other CCT records.

2. Tape directory - this record type exists in the present NASA Landsat-3 tape format and so is included in the design standard; however, as the functions of the tape directory are included in the superstructure, further use of this record type is not recommended.

3. Header - header data are informative in nature. Header records precede the actual data records and generally describe particular data formatting and identification aspects of the data records to follow.

4. Annotation - this record type will contain data that is intended to be displayed along with sensor or experiment data on photographic or graphic products. This type of data may be embedded in the actual data records or may be in records dedicated to annotation data only. This latter record type should be classified as annotation. It may include alphanumeric information to be written on the product as well as other product feature data such as tick marks, borders, latitude and longitude indicators, etc.

5. Ancillary - this classification is broad and includes support data for image data. It may include, for example, computer and experiment command and operation data, ephemeris and attitude data, transformation algorithms, ground control point information, map projection data, etc.

6. Data - for most remote sensing applications this record type will be most numerous as it will generally carry the actual sensor data. The basic data of these records will be actual imagery or other experiment data which may be presented in an array, matrix, or other two-dimensional arrangement; or linear, string or one-dimensional experiment value series; or any data resulting from processing and analysis of experiment data. Data records typically also contain, in record leaders or trailers, support data associated with the data set of the record. It should also be noted that for purposes of photographic film recording, some data records may contain only annotation type data that will be exposed in the film recording process on the film product (e.g., tickmarks, alphanumeric information, borders, etc.), and still be typed data since it is to be treated as though it were image data by the recorder. [It should be noted that although this record type is very broad, it is more specific than the term "data record" which has been used up until this point to indicate any record (other than file descriptor) of a data file.]

7. Trailer - this record type should be used only in the circumstance where additional support data must be recorded after (chronologically and physically in the tape layout) the actual sensor/experiment data. Use of trailer records is discouraged except when system/processing constraints make it necessary.
8. Text - the text record is analogous to the "comment" card convention. It may appear anywhere within a logical volume and does not affect the tape format. It contains alphanumeric (ASCII preferred) information of any type and purpose desired.

It is assumed that every record of a CCT will have as a basic record type a code indicating one of these eight classifications.

7.3.1.2.2 Record Sub-type Codes. The three sub-type codes are used to hierarchically sub-classify record types to finer and finer distinctions, until a complete 4-byte code uniquely identifies a record format. The list of sub-type codes is open-ended; however, new record sub-type codes must be approved. The sub-type codes which have been assigned to date include the following:

First Sub-type Codes

Data Type	Code	Data Type	Code
Volume Descriptor	300)$_8$	File Descriptor	077)$_8$
File Pointer	333)$_8$	Default (no sub-type applicable)	022)$_8$
Image	355)$_8$		
Ground Control Point	011)$_8$	Ephemeris/ Attitude	366)$_8$
Map Projection	044)$_8$	Radiometric Calibration	077)$_8$

Second Sub-type Codes

Data Type	Code
Null (Volume Descriptor)	077)$_8$
Default	022)$_8$

Third Sub-type Codes

Data Type	Code
Default	022)$_8$

It should be noted that the order of the four codes within the overall record type code is: 1) first sub-type, 2) basic record type, 3) second sub-type, 4) third sub-type (see Section 8.3.2). For example, the code for a Landsat-3 image data record would combine the codes: 355)$_8$ (first sub-type, image), 355)$_8$ (basic type, data), 022)$_8$ (second sub-type, default), 022)$_8$ (third sub-type) default) for a 4-byte record code of 355)$_8$ 355)$_8$ 022)$_8$ 022)$_8$.

Similarly, the code for a Landsat-D ancillary record of ephemeris/attitude data would combine: 366)$_8$ (first sub-type, ephemeris/attitude), 044)$_8$ (basic type, ancillary), 022)$_8$ (second sub-type, default), and 022)$_8$ (third sub-type, default) for a 4-byte record code of 366)$_8$ 044)$_8$ 022)$_8$ 022)$_8$.

7.3.2 Record Formatting

In formatting a particular record, there are two types of considerations established by this standard. The first is of basic record construction conventions, and the second is of the formatting of the data contained within the record. These are discussed below.

7.3.2.1 Record Construction. A standard record is a multiple of 180 bytes in length and begins with 12 bytes of introductory data, as illustrated in Figure 7-1. If a record length of a 180-byte multiple is not feasible, the record length should at least be a multiple of four bytes. After the 12-byte record introduction, record format and content depend on record type. The 12 introductory bytes are defined as follows:

- Record Number - a 4-byte binary number which indicates the sequence of the record within the file. The first record of the file is numbered 1, and the record number increments by 1 per record. The record number is binary and is right-justified and the left-most bit is the most significant, as indicated in Section 6.1. These are the first four bytes of the record.

ORIGINAL PAGE IS
OF POOR QUALITY

Byte Number	Context	
1	Record Number	
2		
3		
4		
5	1st Record Sub-type Code	Record Type Code
6	Basic Record Type Code	
7	2nd Record Sub-type Code	
8	3rd Record Sub-type Code	
9	Record Length	
10		
11		
12		
13	Data	
180N		

FIGURE 7-1. A STANDARD CCT RECORD

- Record Type Code - This is comprised of the four 1-byte codes described in Section 7.3.1. The byte asignments are: byte 5, first sub-type code; byte 6, basic type code; byte 7, second sub-type code; byte 8, third sub-type code.

- Record Length - Each standard record has its size in bytes recorded in bytes 9 through 12 for internal verification purposes. This is a binary, right-justified number with the left-most bit most significant, and with the bit numbers and weights assigned to these four bytes in the same manner as for the four bytes of the record number.

7.3.2.2 Data Formatting Consideration. The characteristics of remote sensing data are determined to a great degree by factors which are established previous to tape format design, such as sensor configuration, on-board data handling and ground data processing. These factors present constraints within which the format designer must work, such as a required length in bits per pixel value and required number of pixels and other data grouping. There are, however, some basic guidelines which will be followed whenever possible in designing standard record formats. These are as follows:

- Data groups should be multiples of 8-bit bytes in length. All data fields should be aligned so as to fall on 4-byte boundaries.

- Real and complex numbers may be given only in the formats included in the EIA standard (see Applicable Documents, Section 4).

- Pixel and other data group justification, packing, etc., must be determined so as not to vary, at least within a file and preferably within a logical volume. Codes for describing these factors should be included in the file descriptor variable segment.

- Alphanumeric data should, preferably, be coded in ASCII.

- Data from various sensors may be band-interleaved hy line or pixel, or band-sequential, but when pixels are of differing resolution, they must either be adjusted to a common resolution or else they must be band-sequential.
- Varying pixel length, due to data compression techniques or other causes, should be handled as separate classes of image data files.

7.4 DETERMINING FILE CLASS AND DEFINING THE FILE DESCRIPTOR VARIABLE SEGMENT

The class of a file depends upon its general format, which is dependent on the type and ordering of data within the file. Since the purpose of the file descriptor variable segment is to provide access to the file, files of the same class can be served by the same file descriptor variable segment.

Once the format of a file has been established, which includes defining the type and ordering of records within the file, the first step is to decide whether any of the existing file classes/variable segments are appropriate for the file. Those which exist at present are given in Appendix B. It should be noted that they are presented there in their generic form. When used with a particular format, fields such as "Number of records of group 1 record type" become more specific, e.g., this field may become "Number of header records" or whatever record type is appropriate for the particular file. Also, fields defined in a variable segment may simply not be used for a particular application. Compare the variable segments in Appendix B with those in the example in section 8 for further clarification of this distinction.

If no existing file class/variable segment is appropriate for the new file format, a new class and variable segment must be designated. In doing this, it is desirable to modify existing variable segments to meet new requirements, or at least to pattern new segments on old ones. In any variable segment, the first 36 bytes should be reserved for indication of the number of records of each type in the file. The preferred format includes the following six parameter fields:

- Number of records of group 1 record type (6 bytes)
- Length of group 1 record type records (6 bytes)
- Number of records of group 2 record type (6 bytes)
- Length of group 2 record type records (6 bytes)
- Number of records of group 3 record type (6 bytes)
- Length of group 3 record type records (6 bytes).

These 36 bytes should be reserved even if there are less than three types of records in the file.

When determining what types of fields should be present in a given variable segment, the prime concern is to direct a user to the data considered to be important. In many instances, use of "locators" will serve this purpose. A locator is 16 bytes of information which points to the field in the file which contains the desired information. These 16 bytes are:

- Number of the record containing the field (6 bytes)
- Field location within record, given by the record byte number of the first byte of the field (6 bytes)
- Field size, in bytes (3 bytes)
- Field type; a code for the data type of the field such as:
  - A - alphanumeric in ASCII or EBCDIC
  - B - binary
  - N - numeric in ASCII or EBCDIC.

When a new file descriptor variable segment has been defined, it must be assigned a file class name of 28 characters and a file class code of 4 characters. These will appear in the file pointer records which reference files of this class (see Tables 6-3 and 6-4, field 11).

8.0 BRINGING EXISTING FORMATS INTO THE CCT FAMILY, AN EXAMPLE

The method for adapting existing CCT formats to the CCT family environment, which is illustrated in Figure 5.1, may perhaps be better understood through example. This section will show the modifications required for one such format, that of Landsat-3 CCTs, to be brought into the CCT family of tape formats. This format is specified by Interface Control Document (see Applicable Documents, Section 4) between NASA's Goddard Space Flight Center and the Department of Interior's Earth Resources Observation System Data Center. The Landsat-3 format was chosen because there are a relatively large number of remote sensing data users already familiar with this format, and because the break points for multi-volume sets are established by definition. Since the number of Landsat-3 data records per tape is known, the exact number of superstructure records per volume set can be established.

8.1 THE LANDSAT-3 FORMAT (WITHOUT THE SUPERSTRUCTURE)

Without the superstructure, a Landsat-3 CCT tape set consists of from one to three physical volumes, depending on the type of imagery contained in the set. Each tape set contains one logical volume of data, and each physical volume starts with a tape directory. A logical volume consists of imagery and associated data from one site and time, whether it is a scene of one or more bands of Multi-Spectral Scanner (MSS) imagery or a subscene of Return Beam Vidicon (RBV) imagery. An RBV logical volume has one image data section preceded by a leader section containing scene attributes (headers, ancillary, annotation) and followed by a trailer section. MSS logical volumes will have one image section (with leader and trailer) if only one band (one image) of the scene is present, or if the imagery of the various bands is interleaved by line (the BIL format). If MSS images are separated by band and presented sequentially (the BSQ format), the logical volume will contain as many image sections (each with leader and trailer) as there are bands of imagery present. These three cases are depicted in Figure 8.1.

CCT files are followed by end-of-file (EOF) marks. Tape directory, leader, image, and trailer sections are thus followed by EOFs as shown in Figure 8.2. The number of records per file varies with sensor, interleaving and data correction, but the number of records per file is known for each combination of these factors.

Records of a given tape set are of a constant record length which is established per sensor, except that the tape directory record is always 360 bytes. MSS records are 3596 bytes and RBV records are 5388 bytes. Records are separated by inter-record gaps (IRGs).

The last record of a physical volume is followed by double EOFs; the last record of a tape set is followed by triple EOFs.

8.2 INCORPORATING THE SUPERSTRUCTURE

To bring a particular format into the CCT family of tape formats, essentially all that is required is to insert the proper superstructure records in the proper positions. Two examples of how this is done are given in this section. The first case is of a single-volume set with one image section; the second is of a three-volume set of five image sections.

For the single-volume case, the example format is that for one band of geometrically uncorrected MSS imagery. The superstructure records required are illustrated in Figure 8.3. They are the volume descriptor and four pointer records of the volume directory file, four file descriptor records, and the volume descriptor of the null volume directory file. The additional tape required for these six volume directory records (at 360 bytes each), four file descriptor records (at 3596 bytes), ten IRGs (at a nominal 0.6 inches each) and two EOFs (at a nominal 3.0 inches each) are as follows:

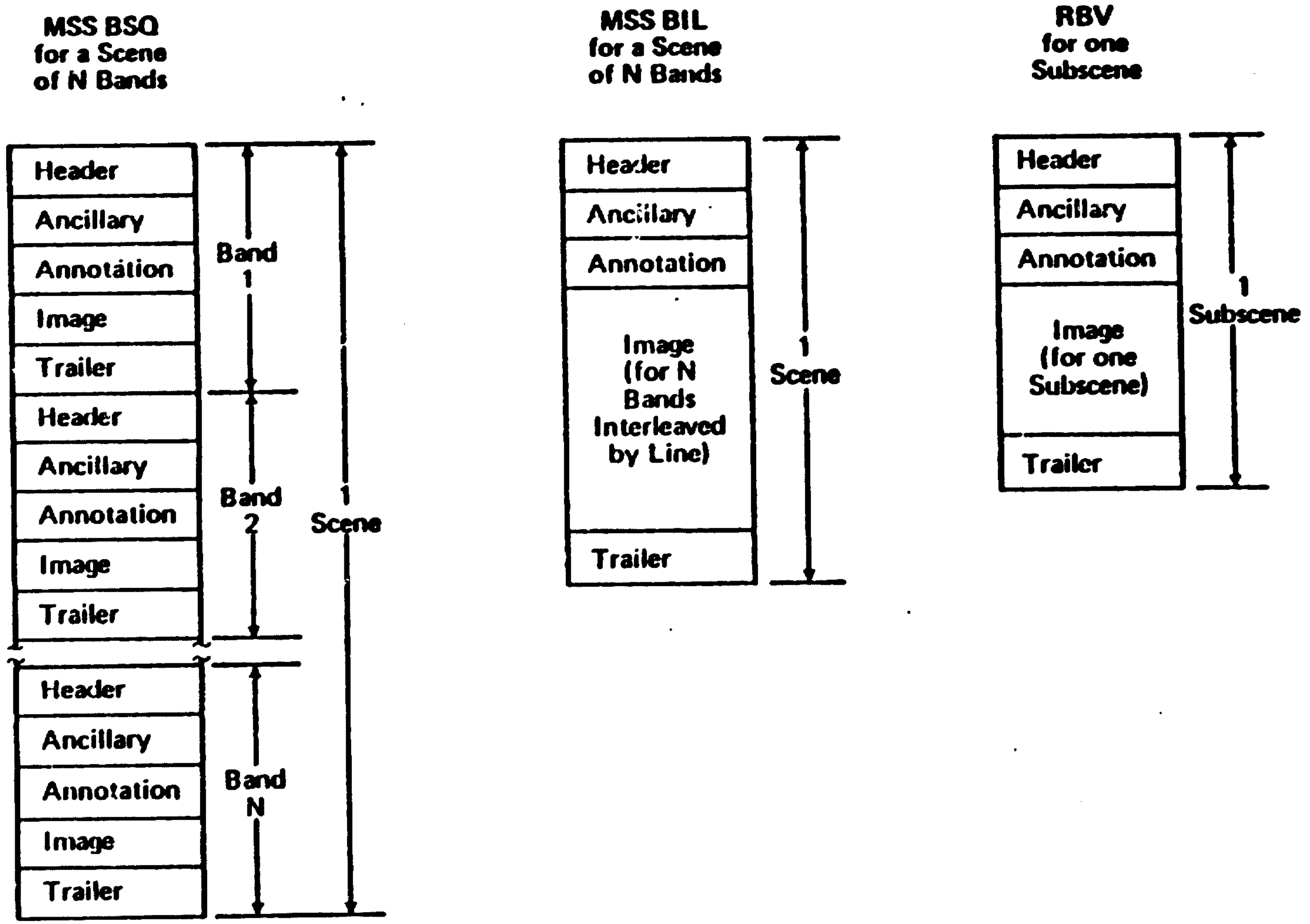


FIGURE 8.1. LOGICAL VOLUME DATA FORMAT OF LANDSAT-3 CCTs (WITHOUT THE SUPERSTRUCTURE)

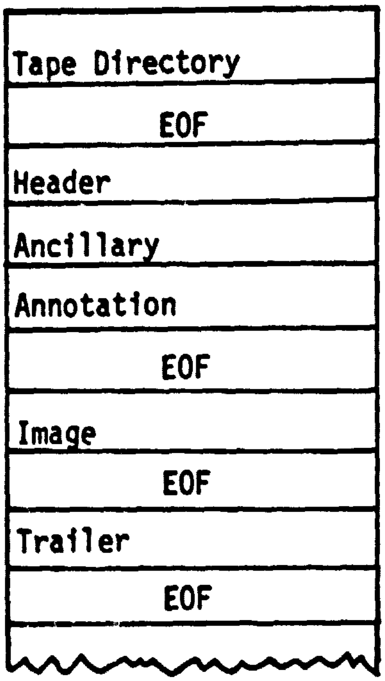


FIGURE 8.2. PLACEMENT OF EOFs

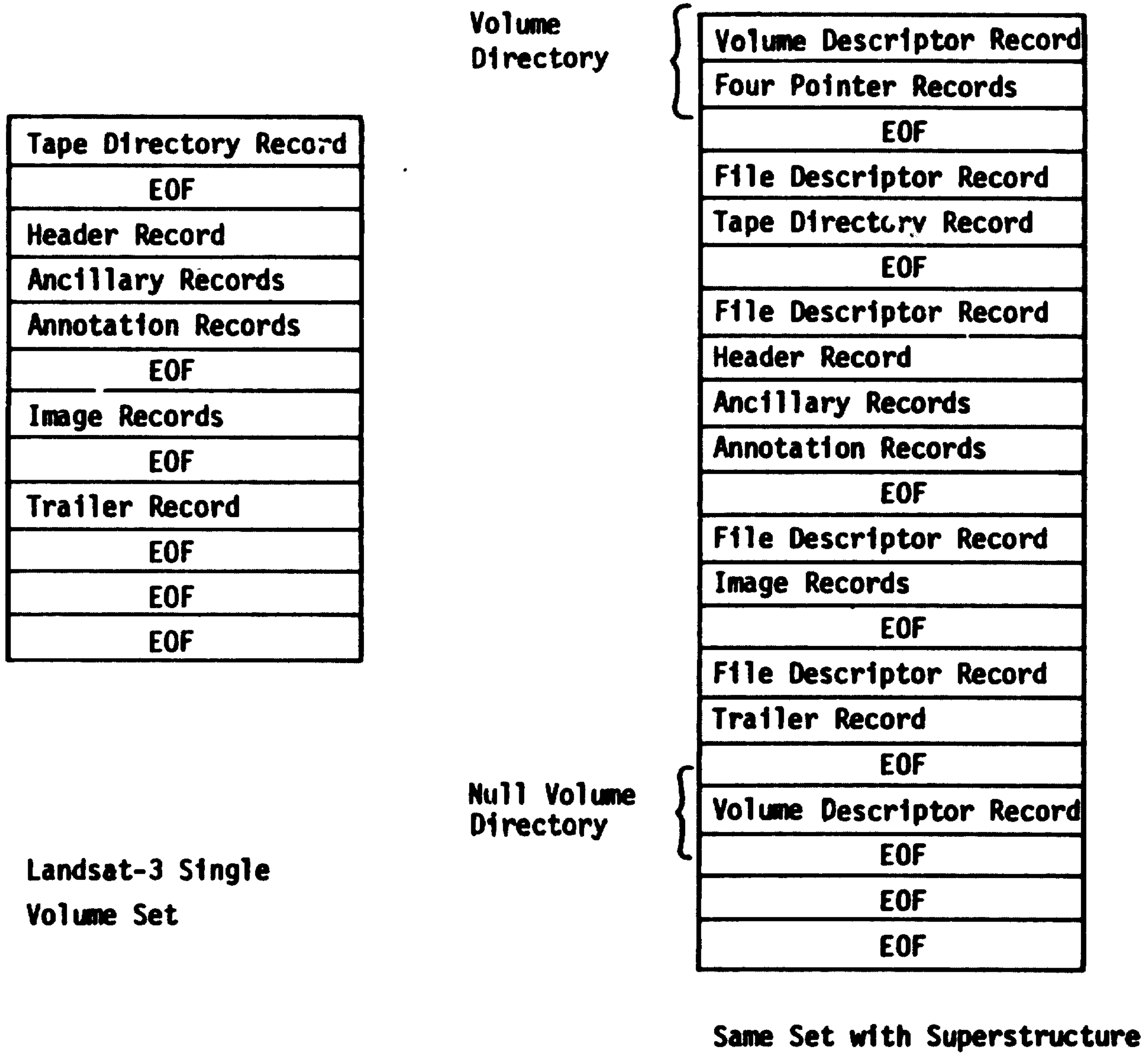


FIGURE 8.3. EXAMPLE OF A SINGLE-VOLUME SET OF ONE BAND OF MSS IMAGERY, BEFORE AND AFTER ADDING SUPERSTRUCTURE RECORDS

$$\frac{4(3596) + 6(360) \text{ bytes}}{800 \text{ bytes/in.}} + 10\ (0.6) \text{ in.} + 2\ (3.0) \text{ in.} = 32.7 \text{ in.}$$

at a recording density of 800 PBI, and

$$\frac{4(3596) + 6(360) \text{ bytes}}{1600 \text{ bytes/in.}} + 10\ (0.6) \text{ in.} + 2\ (3.0) \text{ in.} = 16.3 \text{ in.}$$

at a recording density of 1600 BPI.

When these figures are compared with the 2,400 ft. of tape per CCT, and the approximately 1,020 ft. at 800 BPI and 570 ft. at 1,600 BPI required for the image section alone, the additional tape required for the superstructure appears insignificant.

The example used for the triple volume set format is that of five bands of BSQ, geometrically corrected MSS imagery at 800 BPI (two volumes at 1600 BPI). The placement of superstructure records in this case is illustrated in Figure 8.4. The number of superstructure records required totals 84 (54 for the 1600 BPI case). (It should be noted that when an image section is split between physical volumes each part constitutes a file which is described by its own file descriptor record. This is not standard procedure, but is necessary for Landsat-3 due to the repeated tape directory.) Calculating additonal tape required for the superstructure records in the same manner as above produces the following figures:

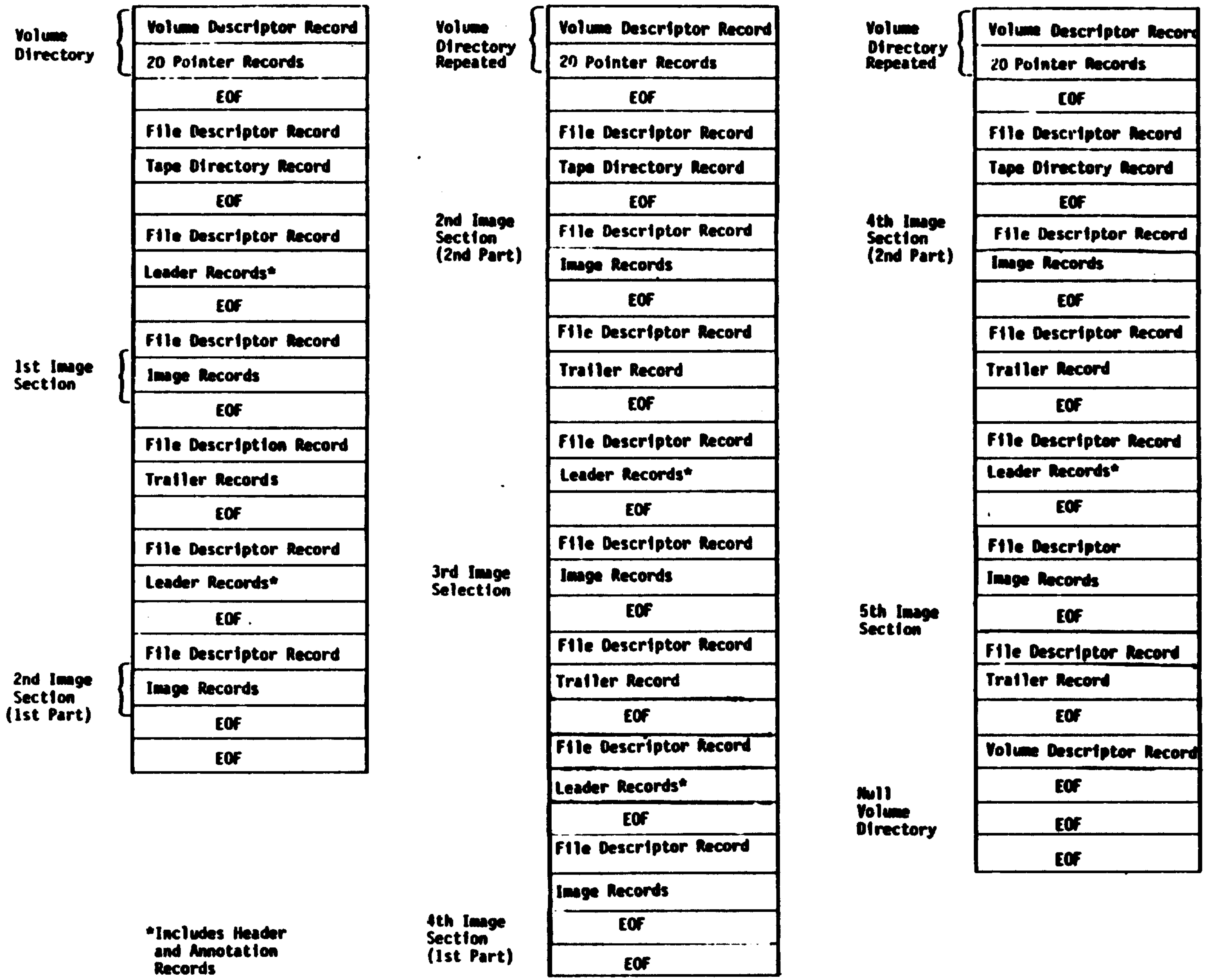
Volume 1, 51.6 in. @ 800 BPI (44.3 in. @ 1600 BPI)
Volume 2, 66.9 in. @ 800 BPI (39.6 in. @ 1600 BPI)
Volume 3, 55.1 in. @ 800 BPI (only 2 tapes required @ 1600 BPI)

Total additional tape required for the tape set, 173.6 in. @ 800 BPI (83.9 in. @ 1600 BPI).

While these numbers are higher per tape than the first example, the 14 1/2 ft. total additional at 800 BPI and 7 ft. total additional at 1600 BPI are still quite insignificant when compared to the total tape required for the five image sections, which is 5660 ft. at 800 BPI and 2870 ft. at 1600 BPI.

The total additional records and tape required for various volume sets of Landsat-3 imagery are given in Table 8.1.

ORIGINAL PAGE IS
OF POOR QUALITY



Volume Directory
- Volume Descriptor Record
- 20 Pointer Records
- EOF
- File Descriptor Record
- Tape Directory Record
- EOF
- File Descriptor Record
- Leader Records*
- EOF

1st Image Section
- File Descriptor Record
- Image Records
- EOF
- File Description Record
- Trailer Records
- EOF
- File Descriptor Record
- Leader Records*
- EOF .

2nd Image Section (1st Part)
- File Descriptor Record
- Image Records
- EOF
- EOF

Volume Directory Repeated
- Volume Descriptor Record
- 20 Pointer Records
- EOF
- File Descriptor Record
- Tape Directory Record
- EOF

2nd Image Section (2nd Part)
- File Descriptor Record
- Image Records
- EOF
- File Descriptor Record
- Trailer Record
- EOF
- File Descriptor Record
- Leader Records*
- EOF

3rd Image Selection
- File Descriptor Record
- Image Records
- EOF
- File Descriptor Record
- Trailer Record
- EOF
- File Descriptor Record
- Leader Records*
- EOF

4th Image Section (1st Part)
- File Descriptor Record
- Image Records
- EOF
- EOF

Volume Directory Repeated
- Volume Descriptor Record
- 20 Pointer Records
- EOF
- File Descriptor Record
- Tape Directory Record
- EOF

4th Image Section (2nd Part)
- File Descriptor Record
- Image Records
- EOF
- File Descriptor Record
- Trailer Record
- EOF
- File Descriptor Record
- Leader Records*
- EOF

5th Image Section
- File Descriptor
- Image Records
- EOF
- File Descriptor Record
- Trailer Record
- EOF

Null Volume Directory
- Volume Descriptor Record
- EOF
- EOF
- EOF

*Includes Header and Annotation Records

FIGURE 8.4. EXAMPLE OF A VOLUME SET OF FIVE BANDS OF MSS IMAGERY AFTER ADDING SUPERSTRUCTURE RECORDS

ORIGINAL PAGE IS OF POOR QUALITY

TABLE 8.1

ADDITIONAL RECORDS AND TAPE FOOTAGE REQUIRED FOR THE SUPERSTRUCTURE, BY VOLUME SET

Date Type and Tape Number	Image Distribution by Density		Number of Additional Records		Additional Tape (ft.) Per Volume Set	
	800 BPI	1600 BPI	800 BPI	1600 BPI	800 BPI	1600 BPI
RBV						
Geometrically Uncorrected						
Tape 1	2062 lines	entire image	10	10	3.45	2.43
Tape 2	remaining image lines		11			
Geometrically Corrected						
Tape 1	2661 lines	entire image	10	10	3.45	2.43
Tape 2	2661 lines		11			
MSS BSQ						
Geometrically Uncorrected						
(1 band) Tape 1	entire image	entire image	10	10	3.15	1.65
(2 bands) Tape 1	band 1	all images	13	16	5.14	2.62
Tape 2	band 2		14			
(3 bands) Tape.1	bands 1 and 2	all images	19	22	6.94	3.54
Tape 2	band 3		17			
(4 bands) Tape 1	bands 1 and 2	all images	22	28	9.16	4.46
Tape 2	bands 3 and 4		23			
(5 bands) Tape 1	bands 1 and 2	bands 1, 2, and 3	26	28	12.71	7.01
Tape 2	bands 3 and 4	bands 4 and 5	26	26		
Tape 3	band 5		24			
Geometrically Corrected						
(1 band) Tape 1	entire image	entire image	10	10	2.39	1.70
(2 bands) Tape 1	band 1 and 1491 lines of band 2	all images	16	16	5.63	2.61
Tape 2	1492 lines of band 2		14			
(3 bands) Tape 1	band 1 and 1491 lines of band 2	all images	19	22	6.59	3.54
Tape 2	1492 lines of band 2 and band 3		20			
(4 bands) Tape 1	band 1 and 1491 lines of band 2	bands 1 and 2	23	22	11.66	5.88
Tape 2	1492 lines of band 2 and band 3	bands 3 and 4	23	23		
Tape 3	band 4		22			
(5 bands) Tape 1	band 1 and 1987 lines of band 2	bands 1, 2, and 3	27	28	13.88	7.01
Tape 2	996 of 2, and band 3, and 996 of 4	bands 4 and 5	29	26		
Tape 3	1987 lines of band 4 and band 5		28			
MSS BIL						
Geometrically Uncorrected						
(4 bands) Tape 1	4800 lines	all lines	9	10	3.75	1.70
Tape 2	4800 lines		9			
(5 bands) Tape 1	4000 lines	6000 lines	10	9	5.44	2.73
Tape 2	4000 lines	6000 lines	8	9		
Tape 3	4000 lines		10			
Geometrically Corrected						
(4 bands) Tape 1	3976 lines	5964 lines	10	9	5.44	2.73
Tape 2	3976 lines	5968 lines	8	9		
Tape 3	3980 lines		10			
(5 bands) Tape 1	4970 lines	7455 lines	10	9	5.44	2.73
Tape 2	4970 lines	7460 lines	8	9		
Tape 3	4975 lines		10			

8.3 ASSIGNING FILE CLASS CODES AND VARIABLE SEGMENT FIELDS

Prerequisite to employing the superstructure is the definition of application-specific codes and fields. A file class and a variable segment must be defined for each different file format used in a particular format. These definitions for Landsat-3 are as follows in this section.

There are four types of files in the Landsat-3 formats. Their names and codes are:

Class Name	Class Code	File Content
TAPE DIRECTORY	TDƀƀ	Tape directory record
LEADER FILE	LEAD	Header, ancillary, and annotation records
IMAGERY FILE	IMGY	Image data records
TRAILER FILE	TRAI	Trailer Record

Each of the four file types has a file descriptor record variable segment format. The variable segment is 180 bytes in length for the tape directory file; it is 3416 bytes for MSS and 5208 for RBV, for all other files. It starts in byte 181 of the file descriptor record. Since the purpose of the tape directory record is superseded by the superstructure, it is assumed that those accessing the tape via superstructure records will skip the tape directory file, and therefore, the tape directory variable segment is simply 180 blanks. The variable segments for the other three files are listed and explained in Tables 8-2 through 8-7.

The field and byte numbers in Tables 8-2 through 8-7 are relative to the variable segment. Field #1 and byte #1 of the variable segment are in fact field #29 and byte #181 of the file descriptor record.

TABLE 8-2.
THE LEADER FILE VARIABLE SEGMENT (LANDSAT-3)

FIELD		Byte #	Description
Type*	Number		
N	1	1-6	Number of header records
N	2	7-12	Header record length
N	3	13-18	Number of ancillary records
N	4	19-24	Ancillary record length
N	5	25-28	Number of annotation records
N	6	29-36	Annotation record length
A	7	37-52	Scene identification field locator
A	8	53-68	World Reference System (WRS) ID field locator
A	9	69-84	Mission identification field locator
A	10	85-100	Sensor identification field locator
A	11	101-116	Exposure date-time field locator
A	12	117-132	Image center field locator
A	13	133-148	Geometric/radiometric correction applied indicator locator
A	14	149-164	Interleaving indicator locator
A	15	165-180	Band(s) indicator locator
A	16	181-196	Subscene indicator locator
	17	197-3416 (MSS) Or 5208 (RBV)	Blanks

* N = numeric, A = alphanumeric

TABLE 8-3.

THE LEADER FILE VARIABLE SEGMENT EXPLAINED (LANDSAT-3)

Field	Explanation
1	The number of header records in the file
2	The length, in bytes, of header records, always 3596 for MSS or 5388 for RBV
3	The number of ancillary records in the file
4	The length, in bytes, of ancillary records, always 3596 for MSS or 5388 for RBV
5	The number of annotation records in the file
6	The length, in bytes, of annotation records, always 3596 for MSS or 5388 for RBV
7-16	While the first six fields of this segment provide actual information, the remaining ten fields are locator fields which point to the position in the file where various information can be found. The location of the desired field is given in 16 bytes coded as follows: 6 bytes - the record sequence number of the record containing the field 6 bytes - the record byte number (referenced from the beginning of the record) of the first byte of the field 3 bytes - length of the field in bytes 1 byte - a code for the type of data in the field. Codes are A = alphanumeric in ASCII or EBCDIC B = binary N = numeric in ASCII or EBCDIC.

TABLE 8-3.

THE LEADER FILE VARIABLE SEGMENT EXPLAINED (LANDSAT-3) Con'd

Field	Explanation
7	Location of the scene identification
8	Location of the WRS identification
9	Location of the mission identification
10	Location of the sensor identification
11	Location of the image exposure date and time field
12	Location of the field which gives the image in latitude/longitude
13	Location of the field which indicates whether radiometric and geometric corrections have been applied to the imagery
14	Location of the field which tells if MSS data is band-interleaved by line or band-sequential
15	Location of the field which indicates which band(s) or image(s) are given in the image section
16	Location of the field which indicates which subscene or subimage is given in the image section.
17	Blanks to fill the record.

TABLE 8-4.
THE IMAGERY FILE VARIABLE SEGMENT (LANDSAT-3)

Field		Byte #	Description
Type*	Number		
N	1	I- 6	Number of image records
N	2	7-12	Image record length
	3	13-36	Reserved (blanks)
			<u>Pixel Group Data</u>
N	4	37-40	Number of bits per pixel
N	5	41-44	Number of pixels per data group
N	6	45-48	Number of bytes per data group
A	7	49-52	Justification and order of pixels within data group
			<u>Image Data in This File</u>
N	8	53-56	Number of images (bands)
N	9	57-64	Number of lines per image (excluding border lines)
N	10	65-68	Number of left border pixels per line
N	11	69-76	Total number of image pixels allocated per line per band (including pad pixels)
N	12	77-80	Number of right border pixels per line
N	13	81-84	Number of top border lines
N	14	85-88	Number of bottom border lines
A	15	89-92	Interleaving indicator
			<u>Record Data in This File</u>
N	16	93-94	Number of physical records per line (coded 0 if < 1)
N	17	95-96	Number of physical records per multispectral line
N	18	97-100	Number of bytes of prefix data per record
N	19	101-108	Number of bytes of image data per record
N	20	109-112	Number of bytes of suffix data per record
A	21	113-116	Prefix/suffix repeat flag

*N = numeric
A = alphanumeric

TABLE 8-4 cont'd

THE IMAGERY FILE VARIABLE SEGMENT (LANDSAT-3)

Field		Byte #	Description
Type	Number		
			Prefix/Suffix Data Locators
A	22	117-124	Scan line number locator
A	23	125-132	Image (band) number locator
A	24	133-140	Time of scan line locator
A	25	141-148	Left-fill count locator
A	26	149-156	Right-fill count locator
A	27	157-188	Blanks
A	28	189-196	Scan line quality code locator
A	29	197-204	Calibration indicator average values locator
A	30	205-212	Gain values field locator
A	31	213-220	Bias values field locator
	32	220-252	Blanks
			Pixel Data Description
N	33	253-256	Number of left-fill bits within pixel
N	34	257-260	Number of right-fill bits within pixels
N	35	261-268	Maximum available data range of pixel starting from zero
	36	269-end of record	Blanks

TABLE 8-5.
THE IMAGERY FILE VARIABLE SEGMENT EXPLAINED (LANDSAT-3)

Field	Explanation	
1	Total numbers of image records in the file	
2	Length of image records, always 3596 for MSS or 5388 for RBV.	
3	A blank field required for consistency in variable segment formats	
4	Always 8	See Appendix B for description
5	Always 1	
6	Always 1	
7	Always blank	
8	The number of images (bands) in this file	
9	Number of lines per image (band)	
10	Always 0, see Appendix B for description	
11	Number of image pixels per line	
12	Always 0	See Appendix B for description
13	Always 0	
14	Always 0	
15	A code indicating MSS data interleaving BSQb̸ = band sequential BILb̸ = band interleaved by line	
16	Number of physical records per line, always = 1	
17	Number of physical records per multispectral line in this file, = 1 for MSS BSQ, 4 for MSS BIL, 1 for RBV.	
18	The length in bytes of the per-line prefix support data field which includes scan line ID, right and left fill count, etc.	
19	The number of bytes of image data per record = 3548 for MSS, 5322 for corrected RBV, 5375 for uncorrected RBV.	

TABLE 8-5 cont'd

THE IMAGERY FILE VARIABLE SEGMENT EXPLAINED (LANDSAT-3)

Field	Explanation
20	The number of bytes of suffix support data (following the image data of a record) such as scan line quality code, gain and bias values, etc.
21	Always blank, see Appendix B for description.
22-31	Prefix/Suffix data locators - While all of the preceding fields give information about the data in the imagery file, the next eleven fields provide locators which point to the location of data within the image records prefix or suffix. The location is given in 8 bytes as follows: 4 bytes - giving the byte number within the prefix or suffix which begins the field to be located 2 bytes - giving the length in bytes of the field to be located. 1 byte - the letter Por S coded in this byte indicates that the information is in the scan line prefix or suffix, respectively. 1 byte - a code indicating the type of data in the field. Codes are: A = alphanumeric in ASCII or EBCDIC B = binary N = numeric in ASCII or EBCDIC
22	Location of the scan line number
23	Location of the image (band) number
24	Location of the time of the scan line
25	Location of the left-fill count
26	Location of the right-fill count
27	A reserved blank field

TABLE 8-5 cont'd

THE IMAGERY FILE VARIABLE SEGMENT EXPLAINED (LANDSAT-3)

Field	Explanation
28	Location of the scan line quality code
29	Location of the calibration indicator and wedge values
30	Location of the gain values field
31	Location of the bias values field
32	Blank
33-35	Pixel Data Description - the following three fields provide specific information concerning the content of data value witnin a pixel
33	Number of left fill bits within each pixel
34	Number of right fill bits within each pixel
35	Maximum available data range of pixel starting fro zero
36	Blanks to fill the record

TABLE 8-6.

THE TRAILER FILE VARIABLE SEGMENT (LANDSAT-3)

FIELD		BYTE #	DESCRIPTION
TYPE*	NUMBER		
N	1	1-4	Number of trailer records
N	2	5-12	Trailer record length
	3	13-36	Reserved (blanks)
A	4	37-52	Parity error count field locator
A	5	53-68	Quality code summary map field locator
	6	69-3416 (MSS) or 5208 (RBV)	Blanks

* N = numeric, A = alphanumeric

TABLE 8-7.

THE TRAILER VARIABLE SEGMENT EXPLAINED (LANDSAT-3)

Field	Explanation
1	The number of trailer records in the file
2	The length, in bytes, of the trailer records(s) always 3596 MSS or 5388 for RBV
3	A blank field required for consistency in variable segment formats
4-5	Fields which give the location of information in the trailer file. The location is given in 16 bytes as described in Table 8-3, fields 7-15.
4	Location of the parity error count
5	Location of the quality code map summary
6	Blanks to fill the record.

APPENDIX A
GLOSSARY

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
Band	A collection of pixels representing a spectral portion of a scene.
BIL	Band Interleaved by Line
Bit	The smallest element of binary, computer-intelligible data.
BPI	Bytes per inch
BSQ	Band Sequential
Byte	A unit of data consisting of eight bits.
CCRS	Canada Centre for Remote Sensing
CCT	Computer-Compatible Tape
CPI	Characters per inch
Data file	When used in conjunction with a superstructure concept, data file refers to the files of a logical volume other than the volume directory file, i.e., the files containing the data for which the tape is being produced.
Data group	The arrangement of data values and pad into a group which is aligned on byte boundaries, and which is repeated throughout the experiment data section of the tape. See Appendix B, Table B-4, fields 4 through 7.

Data record (1) Records of data files other than the file descriptor records.
(2) One of eight basic record types - that carrying the experiment data. See Section 7.3.1.2.

EOF End-of-File marker

EOS End-of-Set marker, consists of 3 EOFs

EOV End-of-Volume marker, consists of 2 EOFs

File class A file class is comprised of a set of file formats which are similar enough to be accessed through one file descriptor variable segment. See section 7.4 and Appendix B.

File descriptor record One of the three superstructure records. A file descriptor record introduces and tells how to read a data file. See Section 6.4.

File descriptor variable segment That segment of the file descriptor record whose format is dependent on file class.

File pointer record One of the three superstructure records. In the volume directory file, which introduces a logical volume, there is one file pointer record for each data file to follow.

Fill pixels In a rectangular image array, fill pixels complete scan lines of fewer image pixels than the array width. Fill pixels have constant, preassigned values (e.g. black or white) while image pixels contain sensor supplied data values. Due to image skew (caused by earth rotation) and other factors, the number of fill pixels varies from scan line to scan line.

Fixed length record A physical record contained in a file in which all of the physical records have the same record size by design.

Group 1 records A general designation of records of the same type within a file.

GSFC Goddard Space Flight Center

Image array	A two-dimensional arrangement of pixels in a series of lines creating a rectangular area which may contain, in addition to actual image pixels, annotation data and fill pixels.
Interleaving	The pattern in which pixels and lines of various bands are arranged (e.g., band interleaved by line or band interleaved by pixel).
IRG	Inter-Record Gap
Left Justified	Technique of positioning data so that the most significant bit appears in the leftmost position.
Locators	A set of data fields found in file descriptor variable segments which contain the information necessary to locate particular information in the remainder of the file. See Appendix B, Table 2, fields 7 through 15; Table 4, fields 22 through 31; and Table 6, fields 4 and 5.
Logical volume	A logical collection of one or more related files recorded consecutively in a volume set. A logical volume may contain one or more than one physical file, but no file may span logical volumes. Logical volumes may span physical volumes, but are wholly contained within a volume set.
LSB	Least significant bit
MSB	Most significant bit
MSS	Multispectral Scanner
Multispectral Line	A full scan line from each spectral band (image).
Multi-volume recording	The recording of a logical volume which spans physical volumes. Techniques for doing this are discussed in Section 5.4.
Multi-volume set	A tape set (volume set, physical volume set) comprised of more than one tape (physical volume).
Null volume directory	A volume directory file consisting of one volume descriptor and indicating the end of a logical volume. See Table 6.1.

Physical record, record A collection of data, in bytes, written to or read from a tape as a unit in a single operation. On tape, records are separated by inter-record gaps (IRGs).

Physical volume, tape A dismountable physical reel of magnetic medium. A physical volume may contain one, more than one, or less than one file. It may not contain partial records. The end of recording on a standard CCT physical volume is indicated by an end-of-volume (EOV) mark.

Physical volume set See tape set.

Pixel One image sensor sample.

Pixel data group A data group whose data values are pixels.

Pointer record See file pointer record.

Prefix data In an image record, the support data which precedes the image data (not to include standard record introductory data such as record number, type and length).

RBV Return Beam Vidicon

Record See physical record.

Right Justified Technique of positioning data so that the least significant bit appears in the rightmost position.

Scan line A full cross track sweep of an active detector (a full scene width).

Scene A delineated site which is spectrally divided into one or more bands, or spacially dividied into subscenes.

Suffix In an image record, the support data which follows the image data.

Superstructure	A combination of three precisely defined records (volume descriptor, file pointer and file descriptor) and a methodology of how to employ them which will allow users to access the data of a tape without requiring knowledge of the particular tape format.
Tape	See physical volume.
Tape set, volume set, physical volume set	A physical collection of one or more tapes (physical volumes).
Variable data segment, variable segment	That segment of the file descriptor record (byte 181 to end of record) whose data fields are variable in that they depend on file class.
Variable length record	A physical record contained in a file in which the physical records may have different record sizes.
Volume set	See tape set.

APPENDIX B
FILE DESCRIPTOR RECORD VARIABLE DATA SEGMENTS

File Class: LEADER FILE   Class Code: LEAD

This class of file precedes image data files, supplying information associated with the image such as image product annotation, ephemeris/attitude data, processing information and other ancillary information.

The leader file variable segment gives the number of each type of record in the file and locates specific data fields within the file. It is listed and explained in Tables B-1 and B-2.

TABLE B-1
THE FILE DESCRIPTOR RECORD VARIABLE SEGMENT FOR THE LEADER FILE

Field		Byte #	Description
Type*	Number		
N	1	1-6	Number of group 1 records
N	2	7-12	Group 1 record length
N	3	13-18	Number of group 2 records
N	4	19-24	Group 2 record length
N	5	25-30	Number of group 3 records
N	6	31-36	Group 3 record length
A	7	37-52	Scene identification field locator
A	8	53-68	World Reference System (WRS) identification locator
A	9	69-84	Mission identification field locator
A	10	85-100	Sensor identification field locator
A	11	101-116	Exposure date-time field locator
A	12	117-132	Geographic reference field locator
A	13	133-148	Image processing performed field locator
A	14	149-164	Imagery format indicator locator
A	15	165-180	Band(s) indicator locator
A	16	181-196	Subscene or subimage indicator locator
	17	197-end of record	Blanks

*N = numeric, A = alphanumeric

TABLE B-2
THE FILE DESCRIPTOR RECORD VARIABLE SEGMENT FOR THE LEADER FILE EXPLAINED

Field	Explanation
1-6	Number of records of each of up to three record types. It is assumed that records of a given type are grouped together. This necessitates that text records be located at the end of the file.
1	The number of group 1 (first record type) records in the file.
2	The length, in bytes, of group 1 records .
3	The number of group 2 records in the file.
4	The length, in bytes, of group 2 records .
5	The number of group 3 records in the file.
6	The length, in bytes, of group 3 records .
7-16	While the first six fields of this variable segment provide actual information, the remaining ten fields are locator fields which point to the position in the file where various information can be found. The location of the desired field is given in 16 bytes coded as follows: 6 bytes - the record number of the record containing the field 6 bytes - the record byte number of the first byte of the field 3 bytes - length of the field in bytes. 1 byte - a code for the type of data in the field. Codes are: A = alphanumeric in ASCII or EBCDIC B = binary N = numeric in ASCII or EBCDIC.
7	Location of the scene identification.
8	Location of the WRS identification.
9	Location of the mission identification.

TABLE B-2 (cont'd)

Field	Explanation
10	Location of the sensor identification.
11	Location of the image exposure date and time field.
12	Location of the field which references the image geographically.
13	Location of the field which indicates what processing has been performed on the image data, e.g., whether radiometric and geometric corrections have been applied to the imagery.
14	Location of the field which tells if the data are band-interleaved by line, or band-sequential, etc.
15	Location of the field which indicates which image(s) or band(s) is (are) given in the image set, and whether the imagery is
16	Location of the field which indicates which subscene or subimage is (are) given in the image set.
17	Blanks to fill the record.

File Class: IMAGERY FILE Class Code: IMGY

This class of file contains the actual image data. It may also contain per-line support data such as quality codes, fill pixel counts, scan line identification, etc.

The imagery file variable data segment gives the number and length of image records; describes the data format in terms of the pixel group, the record content, and the overall image; and gives the location of significant data fields in the record prefix and suffix. It is listed and explained in Tables B-3 and B-4.

TABLE B-3
THE FILE DESCRIPTOR RECORD VARIABLE SEGMENT FOR THE IMAGERY FILE

FIELD			
Type*	Number	Byte #	Description
N	1	1-6	Number of image records
N	2	7-12	Image record length
	3	13-36	Reserved (blanks)
			<u>Pixel Group Data</u>
N	4	37-40	Number of bits per pixel
N	5	41-44	Number of pixels per data group
N	6	45-48	Number of bytes per data group
A	7	49-52	Justification and order of pixels within data group
			<u>Image Data in this File</u>
N	8	53-56	Number of images (bands)
N	9	57-64	Number of lines per image (excluding border lines)
N	10	65-68	Number of left border pixels per line
N	11	69-76	Total number of image pixels allocated per line per band
N	12	77-80	Number of right border pixels per line
N	13	81-84	Number of top border scan lines
N	14	85-88	Number of bottom border scan lines
A	15	89-92	Interleaving indicator
			<u>Record Data in This File</u>
N	16	93-94	Number of physical records per line (coded 0 if <1)
N	17	95-96	Numbers of physical records per multispectral line
N	18	97-100	Length (in bytes) of prefix data per scan line
N	19	101-108	Number of bytes of image data per scan line
N	20	109-112	Length (in bytes) of suffix data per scan line
A	21	113-116	Prefix/suffix repeat flag
			<u>Prefix/Suffix Data Locators</u>
A	22	117-124	Scan line number locator
A	23	125-132	Image (band) number locator
A	24	133-140	Time of scan line locator
A	25	141-148	Laft-fill count locator
A	26	149-156	Right-fill count locator

*N = numeric, A = alphanumeric

TABLE B.3 (cont'd)

FIELD		Byte #	Description
Type*	Number		
	27	157-188	Blanks
A	28	189-196	Scan Line quality code locator
A	29	197-204	Calibration information field locator
A	30	205-212	Gain values field locator
A	31	213-220	Bias values field locator
	32	220-252	Blanks
			<u>Pixel Data Description</u>
N	33	253-256	Number of left-fill bits within pixel
N	34	257-260	Number of right-fill bits within pixels
N	35	261-268	Maximum data range of pixel, (starting from o)
	36	269-end of record	Blanks

TABLE B-4
THE FILE DESCRIPTOR RECORD VARIABLE SEGMENT FOR THE IMAGERY FILE EXPLAINED

Field — **Explanation**

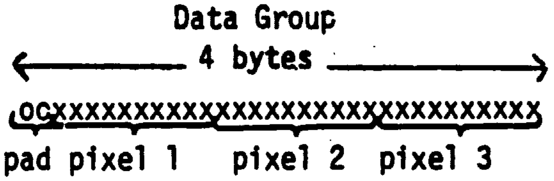
1 Total number of image records in the file.

2 Length of image records.

3 A blank field required for consistency in variable segment formats.

Pixel Group Data

When pixel values are not even byte multiples in length, either each pixel must be padded so as to have the value field length fall on byte boundaries, or the pixels must be grouped so that the group of values plus padding fall on byte boundaries. For example, if each pixel is 10 bits, the pixeT group could be arranged as follows:



The pixel group is then repeated throughout the imagery section. The pixel group of this file is described in the following four fields. (A description of the data value content within each pixel is given in fields 33-35, below.)

4 Number of bits per pixel.

5 Number of pixels per data group.

6 Number of bytes per data group.

7 Justification and order of pixels within data group. This information is given in code. The codes are as follows:

RJLR - the pixels are right justified (i.e., the pad is on the left) with the first pixel leftmost (i.e., pixel order is from left to right).

RJRL - the pixels are right justified with the first pixel rightmost.

LJLR - the pixels are left justified with the first pixel leftmost.

TABLE B-4 (cont'd)

Field — Explanation

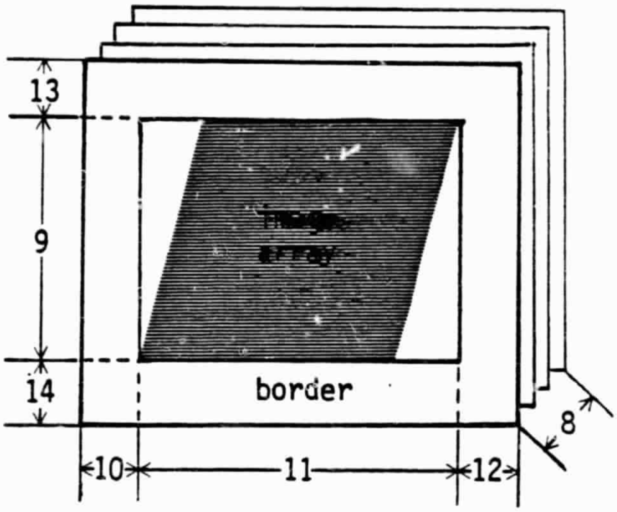
Pixel Group Data

LJRL - the pixels are left justified with the first pixel rightmost.

ƀƀƀƀ - pixel value lengths fall on byte boundaries.

Image Data

The following diagram will be used as an example in explaining the image data parameters. It represents a scene of four bands of geometrically corrected (hence the skew) image data with borders. The numbers indicate the respective fields of the variable segment.



8 The number of images (bands) in this file.

9 Number of lines per image (band) (excluding border pixels).

10 The number of border pixels to the left of the image pixels for each band (image).

11 Total number of image pixels allocated per line or per band.

12 The number of border pixels to the right of the image pixels for each band (image).

13 The number of scan lines per image (band) of border above the image.

14 The number of scan lines per image (band) of border below the image.

15 A code indicating data interleaving.

BSQƀ - band sequential, one (or part of one) scan line per physical record

TABLE B-4 (cont'd)

Field	Explanation
	BSnn - band sequential, nn scan lines per physical record BILƀ - band interleaved by line, one (or part of one) scan line for one band (image) per physical record BInn - band interleaved by line, a maximum of nn bands (images) per physical record, with only the last physical record of the multispectral line being padded on the right BIPƀ - band interleaved by pixel, one multispectral scan line per physical record BIPn - band interleaved by n pixels, one multispectral scan scan line per physical record.
16	Number of physical records per band (image) per scan line (coded 0 if less than 1). This field is not applicable to BIP interleaving, and is coded 0.
17	Number of physical records per multispectral line (BIL) may be less than one for multiple scan lines per physical record (BSnn), in which case it is coded 0. This field is coded 1 for BIPn interleaving.
18	The length of the prefix support data field associated with each scan line of each band (image) which includes scan line ID, right and left fill count, etc. For BIPn interleaving, there is only one prefix data area per physical record.
19	The number of bytes of image data per scan line per band. This includes left and right borders. For BIPn interleaving, it is the total number of bytes of image data within the record.
20	The length of the suffix support data field associated with each scan line of each band (image), such as scan line quality code, gain and bias values, etc. For BIPn interleaving, there is only one suffix data area per physical record.
21	When the scan line requires more than one physical record this field will indicate whether the prefix and suffix fields are repeated in each record. When they are not repeated they are zero-filled to maintain pixel alignment. Rƀƀƀ means they are repeated. ƀƀƀƀ means they are not.
22-31	Prefix/Suffix data locators - While all of the preceding fields give information about the data in the imagery file, the next eleven fields provide locators which point to the location of data within the image record prefix or suffix. The location is given in 8 bytes as follows: 4 bytes - giving the byte number within the prefix or suffix which begins the field to be located. 2 bytes - giving the length in bytes of the field to be located.

TABLE B-4 (cont'd)

Field	Explanation
	1 byte - the letter P or S coded in this byte indicates that the information is in the scan line prefix or suffix, respectively.

	1 byte - a code indicating the type of data in the field. Codes are:
	A = alphanumeric in ASCII or EBCDIC
	B = binary
	N = numeric in ASCII or EBCDIC.

22	Location of the scan line number.
23	Location of the image (band) number.
24	Location of the time of the scan line.
25	Location of the left-fill count.
26	Location of the right-fill count.
27	A reserved blank field.
28	Location of the scan line quality.
29	Location of the calibration values indicator code.
30	Location of the gain values field.
31	Location of the bias values field.
32	Blanks
33-35	Pixel Data Description - the following three fields provide specific information concerning the content of data value within a pixel.
33	Number of left fill bits within each pixel.
34	Number of right fill bits within each pixel.
35	Maximum available data range of pixel starting from zero.
36	Blanks to fill the record.

File Class: TRAILER FILE Class Code: TRAI

This class of file follows image data files, carrying support data which must be recorded after (chronologically and physically in the tape layout) the image data.

The trailer file variable segment gives the number and record length of trailer records, and the location within the file of the parity error count and quality code summary map fields. It is listed and explained in Tables B-5 and B-6.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE B-5
FILE DESCRIPTOR RECORD VARIABLE SEGMENT
FOR THE TRAILER FILE

Field		Byte #	Description
Type*	Number		
N	1	1-4	Number of trailer records
N	2	5-12	Trailer record length
	3	13-36	Reserved (blanks)
A	4-13	37-196	Reserved for locators
	14	197-end of record	Blanks

*N = numeric, A = alphanumeric

TABLE B-6
THE FILE DESCRIPTOR RECORD VARIABLE
SEGMENT FOR THE TRAILER FILE EXPLAINED

Field	Explanation
1	The number of trailer records in the file
2	The length, in bytes, of the trailer record(s)
3	A blank field required for consistency in variable segment formats
4-13	Fields which give the location of information in the trailer file. The location is given in the bytes as described in Table B-2, fields 7-15
14	Blanks to fill the record